SIAM J. Sci. Comput.
Vol. 37, No. 2, pp. A859–A889

A FOURIER COSINE METHOD FOR AN EFFICIENT COMPUTATION OF SOLUTIONS TO BSDEs[*]

M. J. RUIJTER[†] AND C. W. OOSTERLEE[‡]

Abstract. We develop a Fourier method to solve backward stochastic differential equations (BSDEs). A general theta-discretization of the time-integrands leads to an induction scheme with conditional expectations. These are approximated by using Fourier cosine series expansions, relying on the availability of a characteristic function. The method is applied to BSDEs with jumps. Numerical experiments demonstrate the applicability of BSDEs in financial and economic problems and show fast convergence of our efficient probabilistic numerical method.

Key words. backward stochastic differential equations, Fourier cosine expansion method, European options, market imperfections, jump-diffusion process, utility indifference pricing

AMS subject classifications. 91G60, 60H35, 65C30, 65T50, 60E10

DOI. 10.1137/130913183

1. Introduction.

Whereas the theory and applications of classical *forward* stochastic differential equations (FSDEs), with a prescribed initial value, are traditional and have become widely known, we are concerned with *backward* stochastic differential equations (BSDEs). A BSDE is a stochastic differential equation for which a terminal condition, instead of an initial condition, has been specified, and its solution consists of a pair of processes. The linear type of equation was introduced by Bismut in [5], where linear BSDEs were used in stochastic optimal control problems as adjoint equations in the stochastic version of the Pontryagin maximum principle. The general notion of BSDE has been introduced by Pardoux and Peng [37]. They proved existence and uniqueness of solutions of BSDEs under some Lipschitz conditions on the driver function. Many researchers have attempted to relax these restrictions. For example, the authors in [30] show existence of a minimal solution under more general assumption for the driver function, which is assumed to be continuous with linear growth in some of its arguments. Kobylanski [28] provided uniqueness and existence results for a driver with quadratic growth in one of its arguments. For a general introduction to BSDEs we refer the reader to [38, 13].

In recent years, BSDEs have received more attention in mathematical finance and economics. For example, the Black–Scholes formula for pricing options can be represented by a system of decoupled forward-backward stochastic differential equations. Market imperfections can also be incorporated, such as different lending and borrowing rates for money, the presence of transaction costs, or short sales constraints. These imperfections give rise to more involved nonlinear BSDEs. If the asset price follows a jump-diffusion process, then the option cannot be perfectly replicated by assets and cash; i.e., the market is not complete. A way to value and hedge options in this setting is by utility indifference pricing, where a certain utility value is assigned

[*] Submitted to the journal's Methods and Algorithms for Scientific Computing section March 15, 2013; accepted for publication (in revised form) December 9, 2014; published electronically April 16, 2015.
http://www.siam.org/journals/sisc/37-2/91318.html

[†] Centrum Wiskunde & Informatica, Amsterdam, The Netherlands, and CPB Netherlands Bureau for Economic Policy Analysis, Den Haag, The Netherlands (marjonruijter@gmail.com).

[‡] Centrum Wiskunde & Informatica, Amsterdam, The Netherlands, and Delft University of Technology, Delft, The Netherlands (c.w.oosterlee@cwi.nl).

to the possible profits and losses of the hedging portfolio. The pricing problem can be solved by means of a BSDE with jumps.

The well-known Feynman–Kac theorem gives a probabilistic representation for the solution of a linear parabolic partial differential equation (PDE) by means of the corresponding FSDE and a conditional expectation. The solution of a BSDE provides a probabilistic representation for *semilinear parabolic PDEs*; see, for example, [36], which is a generalization of the Feynman–Kac theorem. Also, the converse relation holds. This connection enables us to solve a semilinear PDE by probabilistic numerical methods, like Monte Carlo simulation techniques.

Probabilistic numerical methods to solve BSDEs may, for example, rely on time discretization of the stochastic process and approximations for the appearing conditional expectations. Least-squares Monte Carlo regression to approximate the conditional expectations is used in, for example, [29, 21, 4]. A rich literature exists on other methods, based on, for example, chaos decomposition formulas [11]. In this paper we employ a general theta-method for the time-integration [26] and propose a new method to approximate the solution backward in time. This approach is based on the COS method, which was developed in [16] for pricing financial options. The method is based on Fourier cosine series expansions and relies on the characteristic function of the transitional density, which enables us to approximate the conditional expectations is a very efficient way. The characteristic function is in principle available for Lévy processes or affine jump-diffusion processes. The applicability of the resulting method is therefore quite general. We call the method the *BCOS method*, short for BSDE-COS method.

We start in section 2 with notation, definitions, and a further introduction to BSDEs, where also the link with semilinear PDEs is stated. A general time discretization of the BSDE results in expressions with conditional expectations (section 3). These conditional expectations are computed by the BCOS method (section 4), and the problem is then solved backward in time. We perform extensive numerical experiments in section 5. Then, in section 6, utility indifference pricing and the related maximization problems are discussed. We derive a numerical scheme for the resulting BSDE with jumps in section 6.3. Results in section 7 show the utility indifference ask and bid prices.

2. Backward stochastic differential equations.

We start with some notation and definitions, for which we follow the survey paper [13]. Let $\omega = (\omega_t)_{0\leq t\leq T}$ be a standard one-dimensional Brownian motion on a filtered probability space $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$, with $\mathbb{F} = (\mathcal{F}_t)_{0\leq t\leq T}$ the natural filtration of the Brownian motion ω, and T a fixed finite time horizon. We denote by $\mathbb{H}^2_T(\mathbb{R})$ the set of predictable processes $\eta : \Omega \times [0,T] \to \mathbb{R}$ such that $\mathbb{E}[\int_0^T |\eta_t|^2 dt] < \infty$ and by $\mathbb{L}^2_T(\mathbb{R})$ the set of $\mathcal{F}_T$-measurable random variables $X : \Omega \to \mathbb{R}$ that are square integrable. We consider the BSDE

$$-dY_t = f(t, Y_t, Z_t)dt - Z_t d\omega_t, \quad Y_T = \xi, \tag{2.1}$$

where function $f : \Omega \times [0,T] \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is $\mathcal{P} \otimes \mathcal{B} \otimes \mathcal{B}$-measurable. $\mathcal{P}$ is the set of $\mathcal{F}_t$-progressively measurable scalar processes on $\Omega \times [0,T]$. $f(.)$ is the *generator* or *driver* of the process, and the *terminal condition* $\xi : \Omega \to \mathbb{R}$ is an $\mathcal{F}_T$-measurable random variable. For simplicity we use one-dimensional processes, but the BSDE theory can be extended to higher dimensions, with d-dimensional processes ω_t and Y_t and an $n \times d$-dimensional Z_t process, as described in [13]. A solution to BSDE (2.1) is given by a pair of processes (Y, Z), with Y a continuous real-valued adapted process and Z a real-valued predictable process satisfying $\int_0^T |Z_t|^2 dt < \infty, \mathbb{P}$ a.s., satisfying

$$Y_t = \xi + \int_t^T f(s, Y_s, Z_s)ds - \int_t^T Z_s d\omega_s, \quad 0 \leq t \leq T. \tag{2.2}$$

Unlike an FSDE, the solution of a BSDE is thus a *pair* of adapted processes (Y, Z). Note that BSDEs cannot be considered as time-reversed FSDEs, because at time t the pair (Y_t, Z_t) is $\mathcal{F}_t$-measurable and the process does not yet "know" the terminal condition.

Function f and terminal condition ξ are called *standard parameters* for the BSDE if $\xi \in \mathbb{L}^2_T(\mathbb{R})$, $f(.,0,0) \in \mathbb{H}^2_T(\mathbb{R})$, and f is uniformly Lipschitz in y and z, with Lipschitz constant L_f. A result from [13, 38, 37] is that, given a pair of standard parameters (f, ξ), there exists a unique solution $(Y, Z) \in \mathbb{H}^2_T(\mathbb{R}) \times \mathbb{H}^2_T(\mathbb{R})$ to BSDE (2.1).

Markovian case for the BSDE. A *linear* parabolic PDE has a probabilistic representation by means of the Feynman–Kac theorem. Here, we consider a *semilinear* parabolic PDE of the form

$$-\frac{\partial v}{\partial t}(t,x) - \mathcal{L}v(t,x) - f(t,x,v(t,x),\sigma(t,x)D_x v(t,x)) = 0, \quad (t,x) \in [0,T) \times \mathbb{R}, \tag{2.3a}$$

$$v(T,x) = g(x), \quad x \in \mathbb{R}, \tag{2.3b}$$

with the differential operator of second order

$$\mathcal{L}v(t,x) = \mu(t,x)D_x v(t,x) + \tfrac{1}{2}\sigma^2(t,x)D_x^2 v(t,x). \tag{2.4}$$

This PDE also has a probabilistic representation by means of the FSDE

$$X_t = x, \quad dX_s = \mu(s, X_s)ds + \sigma(s, X_s)d\omega_s, \quad t \leq s \leq T, \tag{2.5}$$

and the BSDE

$$-dY_s = f(s, X_s^{t,x}, Y_s, Z_s)ds - Z_s d\omega_s, \quad Y_T = g(X_T^{t,x}), \tag{2.6}$$

whose terminal condition is determined by the terminal value of FSDE (2.5). $X_s^{t,x}$ denotes the solution to (2.5) starting from x at time t, and $(Y_s^{t,x}, Z_s^{t,x})$ is the corresponding solution to the BSDE.

The coefficients $\sigma : [0,T] \times \mathbb{R} \to \mathbb{R}$ and $\mu : [0,T] \times \mathbb{R} \to \mathbb{R}$ in (2.5) are assumed to be Lipschitz in x and satisfy a linear growth condition in x. Functions $f : [0,T] \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ and $g : \mathbb{R} \to \mathbb{R}$ are assumed to be uniformly continuous with respect to x. Moreover, f satisfies a Lipschitz condition in (y, z), and there exists a constant C such that $|f(t,x,y,z)| + |g(x)| \leq C(1 + |x|^p + |y| + |z|)$, $p \geq 1/2$.

The conditions on f and ξ guarantee the existence of a unique solution (Y, Z) to the BSDE (2.6). Together with the Markov property of the process X, we notice that there exists a deterministic function $v(t,x)$ such that the solution Y of the BSDE is $Y_s^{t,x} = v(s, X_s^{t,x})$, $t \leq s \leq T$. The solution of the BSDE is said to be Markovian as it can be written as a function of time and the state process $X_s^{t,x}$. The following results hold.

RESULT 1 (see [36, 38]). *Let $v \in C^{1,2}$ be a classical solution to* (2.3)*, and suppose there exists a constant $C \geq 0$ such that, for all (t,x), $|v(t,x)| + |\sigma(t,x)D_x v(t,x)| \leq C(1 + |x|)$. Then the pair (Y, Z), defined by*

$$Y_s^{t,x} = v(s, X_s^{t,x}), \quad Z_s^{t,x} = \sigma(s, X_s^{t,x})D_x v(s, X_s^{t,x}), \quad t \leq s \leq T, \tag{2.7}$$

is the solution to BSDE (2.6) *(a so-called verification result).*

The converse result states the following: Suppose (Y, Z) *is the solution to the BSDE; then the function defined by* $v(t, x) = Y_t^{t,x}$ *is a viscosity solution to the PDE.*

The verification result follows from application of Itô's lemma to $v(t, X_t)$ [38]:

$$\begin{aligned} dv(t, X_t) &= (v_t(t, X_t) + \mathcal{L}v(t, X_t))\, dt + \sigma(t, X_t) D_x v(t, X_t) d\omega_t \\ &= -f(t, X_t, v(t, X_t), \sigma(t, X_t) D_x v(t, X_t))dt + \sigma(t, X_t) D_x v(t, X_t) d\omega_t. \end{aligned} \tag{2.8}$$

So, solving the semilinear PDE or the corresponding BSDE results in the same solution. A PDE can be solved by applying numerical discretization techniques, and for BSDEs probabilistic numerical methods are available. For example, Picard methods for Y (see [3, 20]) give rise to a sequence of "easy" linear BSDEs. Another class of methods focuses on dynamic programming equations; see [8, 48, 22, 12]. Our probabilistic solution method to the BSDE is in this class and consists of two steps: First, the FSDE is simulated by a discretization scheme and the general theta-time-discretization of the BSDE then results in expressions with conditional expectations (see section 3). Second, the conditional expectations are computed by the BCOS method (see section 4), and the problem is solved backward in time.

3. Discretization of the BSDE.

We wish to discretize the forward stochastic process,

$$X_0 = x_0 \text{ given}, \quad X_t = X_0 + \int_0^t \mu(s, X_s) ds + \int_0^t \sigma(s, X_s) d\omega_s, \tag{3.1}$$

and the backward process,

$$Y_t = \xi + \int_t^T f(s, \mathbf{X}_s) ds - \int_t^T Z_s d\omega_s, \quad \xi = g(X_T), \tag{3.2}$$

with $\mathbf{X}_s := (X_s, Y_s, Z_s)$. For this, we define a partition $\Delta : 0 = t_0 < t_1 < t_2 < \cdots < t_m < \cdots < t_M = T$, with fixed time steps $\Delta t := t_{m+1} - t_m$. For notational convenience we write $X_m = X_{t_m}$, $Y_m = Y_{t_m}$, $Z_m = Z_{t_m}$, and we define $\Delta\omega_{m+1} := \omega_{t_{m+1}} - \omega_{t_m}$. With ω_t a Wiener process, the increments $\Delta\omega_{m+1} \sim \mathcal{N}(0, \Delta t)$ are normally distributed. The classical Euler discretization X^Δ of the FSDE reads, in this case,

$$X_0^\Delta = x_0, \quad X_{m+1}^\Delta = X_m^\Delta + \mu(t_m, X_m^\Delta)\Delta t + \sigma(t_m, X_m^\Delta)\Delta\omega_{m+1}, \quad m = 0, \ldots, M-1. \tag{3.3}$$

For the BSDE, we then start with

$$Y_m = Y_{m+1} + \int_{t_m}^{t_{m+1}} f(s, \mathbf{X}_s) ds - \int_{t_m}^{t_{m+1}} Z_s d\omega_s. \tag{3.4}$$

By a basic Euler discretization, backward in time, we would require the unknown value Y_{m+1} to approximate Y_m. This scheme hence does not suffice, as it would not take into account the adaptability constraints on Y and Z. To obtain a computationally viable backward induction scheme we should take conditional expectations, which will result in an approximation scheme to the BSDE similar to that used in [51]. For the $\mathcal{F}_{t_m}$-measurable random variables Y_m and Z_m it holds that $\mathbb{E}_m[Y_m] = Y_m$ and

$\mathbb{E}_m[Z_m] = Z_m$, where $\mathbb{E}_m[.]$ represents the conditional expectation $\mathbb{E}[.|\mathcal{F}_{t_m}]$. Taking conditional expectations on both sides of (3.4) then results in

$$\begin{aligned} Y_m &= \mathbb{E}_m[Y_{m+1}] + \int_{t_m}^{t_{m+1}} \mathbb{E}_m[f(s,\mathbf{X}_s)]ds \\ &\approx \mathbb{E}_m[Y_{m+1}] + \Delta t\theta_1 f(t_m,\mathbf{X}_m) + \Delta t(1-\theta_1)\mathbb{E}_m[f(t_{m+1},\mathbf{X}_{m+1})], \quad \theta_1 \in [0,1]. \end{aligned} \tag{3.5}$$

The integrand in (3.5) is a deterministic continuous function of time s, so that we can use the well-known theta-time-discretization method to approximate the integral [26]. Multiplying both sides of (3.4) by $\Delta\omega_{m+1}$, taking the conditional expectation, and applying the theta-method also gives us

$$\begin{aligned} 0 &= \mathbb{E}_m[Y_{m+1}\Delta\omega_{m+1}] + \int_{t_m}^{t_{m+1}} \mathbb{E}_m[f(s,\mathbf{X}_s)(\omega_s - \omega_{t_m})]ds - \int_{t_m}^{t_{m+1}} \mathbb{E}_m[Z_s]ds \\ &\approx \mathbb{E}_m[Y_{m+1}\Delta\omega_{m+1}] + \Delta t(1-\theta_2)\mathbb{E}_m[f(t_{m+1},\mathbf{X}_{m+1})\Delta\omega_{m+1}] \\ &\quad - \Delta t\theta_2 Z_m - \Delta t(1-\theta_2)\mathbb{E}_m[Z_{m+1}], \quad \theta_2 \in [0,1]. \end{aligned} \tag{3.6}$$

Note that for (3.5) and (3.6) we use two different time-discretization parameters θ_1 and θ_2, respectively. The above equations lead to a discrete-time approximation (Y^Δ, Z^Δ) for (Y, Z):

$$Y_M^\Delta = g(X_M^\Delta), \quad Z_M^\Delta = \sigma(t_M, X_M^\Delta)D_x g(X_M^\Delta), \tag{3.7a}$$

for $m = M-1, \ldots, 0:$

$$\begin{aligned} Z_m^\Delta &= -\theta_2^{-1}(1-\theta_2)\mathbb{E}_m[Z_{m+1}^\Delta] + \tfrac{1}{\Delta t}\theta_2^{-1}\mathbb{E}_m[Y_{m+1}^\Delta\Delta\omega_{m+1}] \\ &\quad + \theta_2^{-1}(1-\theta_2)\mathbb{E}_m[f(t_{m+1},\mathbf{X}_{m+1}^\Delta)\Delta\omega_{m+1}], \end{aligned} \tag{3.7b}$$

$$Y_m^\Delta = \mathbb{E}_m[Y_{m+1}^\Delta] + \Delta t\theta_1 f(t_m,\mathbf{X}_m^\Delta) + \Delta t(1-\theta_1)\mathbb{E}_m[f(t_{m+1},\mathbf{X}_{m+1}^\Delta)]. \tag{3.7c}$$

The use of $\theta_1 = 0$ gives us an *explicit* scheme for Y_m^Δ, whereas $\theta_1 \in (0,1]$ results in an *implicit* scheme. To solve for Z_m^Δ, we should obviously have $\theta_2 \neq 0$ in (3.7b), which gives an explicit scheme for Z_m^Δ. For the terminal value Z_M^Δ we use the relation from Result 1. At the points where g is not continuously differentiable, we consider a one-sided derivative.[1]

The terminal condition is a deterministic function of X_M^Δ, and X^Δ is a Markov process. Then it is easily seen, using an induction argument, that there are deterministic functions $y(t_m, x)$ and $z(t_m, x)$ so that

$$Y_m^\Delta = y(t_m, X_m^\Delta), \quad Z_m^\Delta = z(t_m, X_m^\Delta). \tag{3.8}$$

So, the random variables Y_m^Δ and Z_m^Δ are functions of X_m^Δ, and the conditional expectations can be replaced by $\mathbb{E}_m^x[.] \equiv \mathbb{E}[.|X_m^\Delta = x]$. Note that functions y and z depend on the discretization partition Δ.

Equations (3.7) provide us with a scheme to solve the BSDE backward in time, starting at terminal time T. One could use least-squares Monte Carlo methods, like the Longstaff–Schwartz method, to approximate the conditional expectations; see, for example, [29, 21, 4]. The authors of [8] apply a Malliavin-based algorithm to solve them, whereas [32] employs a binomial tree method. In the next section, we introduce a Fourier method to solve the BSDE.

[1] It is also possible to take $\theta_1 = \theta_2 = 1$ in the first iteration with time step $(\Delta t)^2$, which gives the same convergence results.

4. BCOS method.

In this section we explain our method of choice to compute the conditional expectations in (3.7) and solve the problem recursively, backward in time. Our method is an extension of the COS method, which is a Fourier method developed in [16] to compute European option prices. The COS method for computing Bermudan options also consists of a backward-in-time scheme to find the conditional expectations of the continuation value; see [17]. The method for solving BSDEs with a COS method is called the *BCOS method* here. First, in section 4.1 we derive the COS formulas and define the Fourier cosine coefficients. Then, sections 4.2 and 4.3 are devoted to the approximation of functions z and y. Section 4.4 discusses the recursive recovery of the Fourier coefficients, and section 4.5 the error components.

4.1. COS formulas and Fourier cosine coefficients.

Suppose we wish to approximate the expectation

$$I := \mathbb{E}_m^x[v(t_{m+1}, X_{m+1}^\Delta)] = \int_{\mathbb{R}} v(t_{m+1}, \zeta) p(\zeta|x) d\zeta, \tag{4.1}$$

where v represents a general functional and $p(\zeta|x) = \mathbb{P}(X_{m+1}^\Delta = \zeta | X_m^\Delta = x)$ denotes the continuous transitional density function. We assume that the integrand decays to zero as $\zeta \to \pm\infty$. Because of that, we can truncate the infinite integration range to a sufficiently finite interval $[a, b] \subset \mathbb{R}$ without losing significant mass of the density. This gives us the approximation

$$I_1 = \int_a^b v(t_{m+1}, \zeta) p(\zeta|x) d\zeta. \tag{4.2}$$

The notation I_i is used to denote the different approximations of I and keeps track of the numerical errors that set in at each step. Next, we replace function v by its Fourier cosine series expansions on $[a, b]$, that is,

$$v(t_{m+1}, \zeta) = {\sum_{k=0}^{\infty}}' \mathcal{V}_k(t_{m+1}) \cos\left(k\pi \frac{\zeta - a}{b - a}\right), \tag{4.3}$$

with series coefficients $\{\mathcal{V}_k\}_{k=0}^\infty$ given by

$$\mathcal{V}_k(t_{m+1}) := \frac{2}{b - a} \int_a^b v(t_{m+1}, \zeta) \cos\left(k\pi \frac{\zeta - a}{b - a}\right) d\zeta. \tag{4.4}$$

$\sum'$ indicates that the first term in the summation is weighted by one-half. We interchange summation and integration and define

$$\mathcal{P}_k(x) := \frac{2}{b - a} \int_a^b p(\zeta|x) \cos\left(k\pi \frac{\zeta - a}{b - a}\right) d\zeta, \tag{4.5}$$

which are the Fourier cosine series coefficients of the transitional density function $p(\zeta|x)$ on $[a, b]$, i.e.,

$$p(\zeta|x) = {\sum_{k=0}^{\infty}}' \mathcal{P}_k(x) \cos\left(k\pi \frac{\zeta - a}{b - a}\right). \tag{4.6}$$

Truncation of the series summations gives us the approximation

$$I_2 = \frac{b - a}{2} {\sum_{k=0}^{N-1}}' \mathcal{V}_k(t_{m+1}) \mathcal{P}_k(x). \tag{4.7}$$

The Fourier cosine coefficients of the transitional density function can now be approximated as follows [16]:

$$\begin{aligned}\mathcal{P}_k(x) &\approx \frac{2}{b-a}\int_{\mathbb{R}} p(\zeta|x)\cos\left(k\pi\frac{\zeta-a}{b-a}\right)d\zeta = \frac{2}{b-a}\Re\left(\varphi\left(\frac{k\pi}{b-a}\Big|x\right)e^{ik\pi\frac{-a}{b-a}}\right)\\ &= \frac{2}{b-a}\Re\left(\phi\left(\frac{k\pi}{b-a}\Big|x\right)e^{ik\pi\frac{x-a}{b-a}}\right) := \Phi_k(x). \qquad (4.8)\end{aligned}$$

$\Re(.)$ denotes taking the real part of the input argument, and $\varphi(.|x)$ is the conditional *characteristic function* of X^{Δ}_{m+1}, given $X^{\Delta}_m = x$. The characteristic function encountered here can be written as

(4.9)

$$\varphi(u|x) = \varphi(u|0)e^{iux} = \phi(u|x)e^{iux}, \quad \phi(u|x) := \exp\left(iu\mu(t_m,x)\Delta t - \tfrac{1}{2}u^2\sigma^2(t_m,x)\Delta t\right).$$

Inserting the above equations into (4.7) gives us the *COS formula* for approximation of I:

(4.10)

$$\hat{I} := I_3 = {\sum_{k=0}^{N-1}}{}' \mathcal{V}_k(t_{m+1})\Re\left(\phi\left(\frac{k\pi}{b-a}\Big|x\right)e^{ik\pi\frac{x-a}{b-a}}\right) = \frac{b-a}{2}{\sum_{k=0}^{N-1}}{}' \mathcal{V}_k(t_{m+1})\Phi_k(x).$$

In order to solve the BSDE, we need to deal with expectations of the form $\mathbb{E}^x_m[v(t_{m+1}, X^{\Delta}_{m+1})\Delta\omega_{m+1}]$. With the help of the equality (A.3) in Appendix A.1, they can be computed by

$$\begin{aligned}\mathbb{E}^x_m[v(t_{m+1}, X^{\Delta}_{m+1})\Delta\omega_{m+1}] &\approx {\sum_{k=0}^{N-1}}{}' \mathcal{V}_k(t_{m+1})\mathbb{E}^x_m\left[\cos\left(k\pi\frac{X^{\Delta}_{m+1}-a}{b-a}\right)\Delta\omega_{m+1}\right]\\ &= \sigma(t_m,x)\Delta t{\sum_{k=0}^{N-1}}{}' \mathcal{V}_k(t_{m+1})\mathbb{E}^x_m\left[\frac{-k\pi}{b-a}\sin\left(k\pi\frac{X^{\Delta}_{m+1}-a}{b-a}\right)\right]\\ &\approx \sigma(t_m,x)\Delta t{\sum_{k=0}^{N-1}}{}' \mathcal{V}_k(t_{m+1})\Re\left(i\frac{k\pi}{b-a}\phi\left(\frac{k\pi}{b-a}\Big|x\right)e^{ik\pi\frac{x-a}{b-a}}\right)\\ &:= \sigma(t_m,x)\Delta t\frac{b-a}{2}{\sum_{k=0}^{N-1}}{}' \mathcal{V}_k(t_{m+1})\Phi'_k(x). \qquad (4.11)\end{aligned}$$

Now we return to the BSDE problem (3.7), where we defined the deterministic functions $y(t_m, X^{\Delta}_m) = Y^{\Delta}_m$ and $z(t_m, X^{\Delta}_m) = Z^{\Delta}_m$. Let $\mathcal{Y}_k(t_{m+1})$ be the Fourier cosine coefficients of $y(t_{m+1}, x)$ in (3.7c), i.e.,

$$\mathcal{Y}_k(t_{m+1}) = \frac{2}{b-a}\int_a^b y(t_{m+1},x)\cos\left(k\pi\frac{x-a}{b-a}\right)dx, \qquad (4.12)$$

$\mathcal{Z}_k(t_{m+1})$ the Fourier cosine coefficients of function $z(t_{m+1}, x)$ in (3.7b), i.e.,

$$\mathcal{Z}_k(t_{m+1}) = \frac{2}{b-a}\int_a^b z(t_{m+1},x)\cos\left(k\pi\frac{x-a}{b-a}\right)dx, \qquad (4.13)$$

and $\mathcal{F}_k(t_{m+1})$ the Fourier cosine coefficients of driver $f(t_{m+1}, x, y(t_{m+1}, x), z(t_{m+1}, x))$, i.e.,

$$\mathcal{F}_k(t_{m+1}) = \frac{2}{b-a} \int_a^b f(t_{m+1}, x, y(t_{m+1}, x), z(t_{m+1}, x)) \cos\left(k\pi \frac{x-a}{b-a}\right) dx. \tag{4.14}$$

In sections 4.2 and 4.3, we will assume that the above coefficients are given. In section 4.4 the algorithm to recover these coefficients recursively, backward in time, will be discussed.

4.2. COS approximation of function $z(t_m, x)$. For the computation of $z(t_m, x)$ in (3.7b), we need to compute three expectations, $\mathbb{E}_m^x[Z_{m+1}^\Delta]$, $\mathbb{E}_m^x[Y_{m+1}^\Delta \Delta\omega_{m+1}]$, and $\mathbb{E}_m^x[f(t_{m+1}, \mathbf{X}_{m+1}^\Delta)\Delta\omega_{m+1}]$. With the help of COS formulas we can derive the following approximations for these expectations:

(4.15a)

$$\mathbb{E}_m^x[Z_{m+1}^\Delta] \approx {\sum_{k=0}^{N-1}}' \mathcal{Z}_k(t_{m+1}) \Re\left(\phi\left(\frac{k\pi}{b-a}\Big| x\right) e^{ik\pi\frac{x-a}{b-a}}\right),$$

(4.15b)

$$\mathbb{E}_m^x[Y_{m+1}^\Delta \Delta\omega_{m+1}] \approx {\sum_{k=0}^{N-1}}' \mathcal{Y}_k(t_{m+1})\sigma(t_m, x)\Delta t \Re\left(\frac{ik\pi}{b-a}\phi\left(\frac{k\pi}{b-a}\Big| x\right) e^{ik\pi\frac{x-a}{b-a}}\right),$$

(4.15c)

$$\mathbb{E}_m^x[f(t_{m+1}, \mathbf{X}_{m+1}^\Delta)\Delta\omega_{m+1}] \approx {\sum_{k=0}^{N-1}}' \mathcal{F}_k(t_{m+1})\sigma(t_m, x)\Delta t \Re\left(\frac{ik\pi}{b-a}\phi\left(\frac{k\pi}{b-a}\Big| x\right) e^{ik\pi\frac{x-a}{b-a}}\right).$$

We then find as COS approximation

$$\begin{aligned} z(t_m, x) \approx & -\frac{1-\theta_2}{\theta_2}\frac{b-a}{2} {\sum_{k=0}^{N-1}}' \mathcal{Z}_k(t_{m+1})\Phi_k(x) \\ & + \frac{b-a}{2} {\sum_{k=0}^{N-1}}' \left(\frac{1}{\Delta t\theta_2}\mathcal{Y}_k(t_{m+1}) + \frac{1-\theta_2}{\theta_2}\mathcal{F}_k(t_{m+1})\right)\sigma(t_m, x)\Delta t \Phi_k'(x), \end{aligned} \tag{4.16}$$

with Φ_k and Φ_k' as defined in (4.8) and (4.11), respectively.

4.3. COS approximation of function $y(t_m, x)$. For the computation of function $y(t_m, x)$ in (3.7c) there are two explicit parts, $\mathbb{E}_m^x[Y_{m+1}^\Delta]$ and $\mathbb{E}_m^x[f(t_{m+1}, \mathbf{X}_{m+1}^\Delta)]$, that are approximated by the following COS formulas:

$$\mathbb{E}_m^x[Y_{m+1}^\Delta] \approx {\sum_{k=0}^{N-1}}' \mathcal{Y}_k(t_{m+1}) \Re\left(\phi\left(\frac{k\pi}{b-a}\right) e^{ik\pi\frac{x-a}{b-a}}\right), \tag{4.17a}$$

$$\mathbb{E}_m^x[f(t_{m+1}, \mathbf{X}_{m+1}^\Delta)] \approx {\sum_{k=0}^{N-1}}' \mathcal{F}_k(t_{m+1}) \Re\left(\phi\left(\frac{k\pi}{b-a}\right) e^{ik\pi\frac{x-a}{b-a}}\right). \tag{4.17b}$$

In addition, when $\theta_1 > 0$, we also have an implicit part, for which we define

$$h(t_m, x) := \mathbb{E}_m^x[Y_{m+1}^\Delta] + \Delta t(1-\theta_1)\mathbb{E}_m^x[f(t_{m+1}, \mathbf{X}_{m+1}^\Delta)]$$

$$\approx \frac{b-a}{2}\sum_{k=0}^{N-1}{}' \mathcal{Y}_k(t_{m+1})\Phi_k(x) + \Delta t(1-\theta_1)\frac{b-a}{2}\sum_{k=0}^{N-1}{}' \mathcal{F}_k(t_{m+1})\Phi_k(x), \tag{4.18}$$

with Φ_k from (4.8). Now we can write

$$y(t_m, x) = \Delta t\theta_1 f(t_m, x, y(t_m, x), z(t_m, x)) + h(t_m, x). \tag{4.19}$$

In order to determine function $y(t_m, x)$ in (4.19), we will perform P *Picard iterations* (see also [21]), starting with an initial guess, $y^0(t_m, x) := \mathbb{E}_m^x[Y_{m+1}^\Delta]$ (see (4.17a)). The convergence properties of the Picard iterations to the "true" values $y(t_m, x)$ are discussed in section 4.5.

4.4. Recovery of coefficients and algorithm. The computation of functions $z(t_m, x)$ and $y(t_m, x)$ at time-point t_m requires the Fourier cosine coefficients $\mathcal{Z}_k(t_{m+1})$, $\mathcal{Y}_k(t_{m+1})$, and $\mathcal{F}_k(t_{m+1})$ at time-point t_{m+1}. For the next time step in the BCOS method we wish to compute functions $z(t_{m-1}, x)$ and $y(t_{m-1}, x)$ at time-point t_{m-1}, for which we need the Fourier cosine coefficients of time-point t_m. The coefficients can be computed recursively, backward in time, as we explain in this section.

We assume a constant drift μ and volatility σ here, and

$$X_{m+1}^\Delta = X_m^\Delta + \mu\Delta t + \sigma\Delta\omega_{m+1}. \tag{4.20}$$

Now, function $\phi(u)$ does not depend on x. In Remark 1 we will comment on the use of more general functions $\mu(t, x)$ and $\sigma(t, x)$.

First, the computation of the coefficients

$$\mathcal{Z}_k(t_m) = \frac{2}{b-a}\int_a^b z(t_m, x)\cos\left(k\pi\frac{x-a}{b-a}\right)dx \tag{4.21}$$

can be divided into three parts, similar to (4.15). We then use the approximations

$$\frac{2}{b-a}\int_a^b \mathbb{E}_m^x[Z_{m+1}^\Delta]\cos\left(k\pi\frac{x-a}{b-a}\right)dx \approx \Re\left(\sum_{j=0}^{N-1}{}' \mathcal{Z}_j(t_{m+1})\phi\left(\frac{j\pi}{b-a}\right)\mathcal{M}_{k,j}\right), \tag{4.22a}$$

$$\frac{2}{b-a}\int_a^b \mathbb{E}_m^x[Y_{m+1}^\Delta\Delta\omega_{m+1}]\cos\left(k\pi\frac{x-a}{b-a}\right)dx$$

$$\approx \Re\left(\sum_{j=0}^{N-1}{}' \frac{ij\pi}{b-a}\sigma\Delta t\mathcal{Y}_j(t_{m+1})\phi\left(\frac{j\pi}{b-a}\right)\mathcal{M}_{k,j}\right), \tag{4.22b}$$

$$\frac{2}{b-a}\int_a^b \mathbb{E}_m^x[f(t_{m+1}, \mathbf{X}_{m+1}^\Delta)\Delta\omega_{m+1}]\cos\left(k\pi\frac{x-a}{b-a}\right)dx$$

$$\approx \Re\left(\sum_{j=0}^{N-1}{}' \frac{ij\pi}{b-a}\sigma\Delta t\mathcal{F}_j(t_{m+1})\phi\left(\frac{j\pi}{b-a}\right)\mathcal{M}_{k,j}\right), \tag{4.22c}$$

with matrix elements

$$\mathcal{M}_{k,j} := \frac{2}{b-a}\int_a^b e^{ij\pi\frac{x-a}{b-a}}\cos\left(k\pi\frac{x-a}{b-a}\right)dx. \tag{4.23}$$

These approximations can be found by inserting COS formulas (4.15). Note that the approximation signs "≈" are due to the errors of the COS formulas, i.e., truncation of the integration range to a finite interval $[a, b]$, truncation of the infinite sums to a finite number of terms N, and the substitution of the series coefficients by the characteristic function approximation. The coefficients $\mathcal{Z}_k(t_m)$ are then computed as follows:

$$\begin{aligned}\mathcal{Z}_k(t_m) \approx \Re\Bigg(&\sum_{j=0}^{N-1}{}' \Bigg[-\frac{1-\theta_2}{\theta_2}\mathcal{Z}_j(t_{m+1})\\ &+\frac{ij\pi}{b-a}\sigma\Delta t\left(\frac{1}{\Delta t\theta_2}\mathcal{Y}_j(t_{m+1})+\frac{1-\theta_2}{\theta_2}\mathcal{F}_j(t_{m+1})\right)\Bigg]\phi\left(\frac{j\pi}{b-a}\right)\mathcal{M}_{k,j}\Bigg).\end{aligned} \tag{4.24}$$

Second, the coefficients $\mathcal{H}_k(t_m)$ of function $h(t_m, x)$ in (4.18) are computed by

$$\begin{aligned}\mathcal{H}_k(t_m) &= \frac{2}{b-a}\int_a^b h(t_m,x)\cos\left(k\pi\frac{x-a}{b-a}\right)dx\\ &\approx \Re\left(\sum_{j=0}^{N-1}{}' \left[\mathcal{Y}_j(t_{m+1})+\Delta t(1-\theta_1)\mathcal{F}_j(t_{m+1})\right]\phi\left(\frac{j\pi}{b-a}\right)\mathcal{M}_{k,j}\right).\end{aligned} \tag{4.25}$$

The Fourier cosine coefficients $\mathcal{Z}_k(t_m)$ in (4.24) and $\mathcal{H}_k(t_m)$ in (4.25), for $k = 0, 1, \ldots, N-1$, can thus be computed by one matrix-vector multiplication. These matrix-vector multiplications $\mathcal{M}\mathbf{u}$ can be done efficiently with the use of an FFT algorithm; see [17]. With this the computational complexity is reduced from order $\mathcal{O}(N^2)$ to order $\mathcal{O}(N \log N)$, with N the number of terms in the summations.

Finally, the coefficients $\mathcal{F}_k^{P-1}(t_m)$ of function $f(t_m, x, y^{P-1}(t_m, x), z(t_m, x))$ are given by

$$\mathcal{F}_k^{P-1}(t_m) := \frac{2}{b-a}\int_a^b f(t_m,x,y^{P-1}(t_m,x),z(t_m,x))\cos\left(k\pi\frac{x-a}{b-a}\right)dx. \tag{4.26}$$

They are approximated by a *discrete Fourier cosine transform* (DCT). For this we need to compute the integrand $f(t_m, x, y^{P-1}(t_m, x), z(t_m, x))$ on an *equidistant x-grid* with N grid points, as explained in the supplementary material [43]. With a converging Picard method, we have $\mathcal{F}_k(t_m) \approx \mathcal{F}_k^{P-1}(t_m)$ for sufficiently many iterations P. Then,

$$\mathcal{Y}_k(t_m) \approx \Delta t\theta_1\mathcal{F}_k^{P-1}(t_m) + \mathcal{H}_k(t_m). \tag{4.27}$$

With the aforementioned formulas we approximate the Fourier cosine coefficients $\mathcal{Z}_k(t_m)$, $\mathcal{Y}_k(t_m)$, and $\mathcal{F}_k(t_m)$ by using the coefficients of time-point t_{m+1}. Starting with the coefficients at the terminal time, we can solve them recursively, backward in time. The evolution of the extra error introduced by approximation of the coefficients has been discussed in detail in [17]. The final approximations of the functions $y(t_m, x)$ and $z(t_m, x)$ by the BCOS method are denoted by $\hat{y}(t_m, x)$ and $\hat{z}(t_m, x)$, respectively.

The overall algorithm to solve the BSDE (3.7) backward in time can be summarized as follows.

Algorithm 1. *(BCOS method)*
Initial step: *Compute, or approximate, the terminal coefficients* $\mathcal{Y}_k(t_M)$, $\mathcal{Z}_k(t_M)$, *and* $\mathcal{F}_k(t_M)$.
Loop: *For* $m = M-1$ *to* $m = 1$*: Compute the functions* $\hat{z}(t_m, x)$, $f(t_m, x, \hat{y}(t_m, x), \hat{z}(t_m, x))$, *and* $\hat{y}(t_m, x)$, *and determine the corresponding Fourier cosine coefficients* $\mathcal{Z}_k(t_m)$, $\mathcal{F}_k(t_m)$, *and* $\mathcal{Y}_k(t_m)$, *as described in sections* 4.2, 4.3, *and* 4.4.
Terminal step: *Compute* $\hat{z}(t_0, x_0)$ *and* $\hat{y}(t_0, x_0)$.

Remark 1. For general drift $\mu(t,x)$ and volatility $\sigma(t,x)$ in (4.20) we need to compute the following integrals to recover the Fourier cosine coefficients:

$$\frac{2}{b-a}\int_a^b \phi\left(\frac{j\pi}{b-a}\Big| x\right) e^{ij\pi\frac{x-a}{b-a}} \cos\left(k\pi\frac{x-a}{b-a}\right) dx, \tag{4.28}$$

which is not equal to $\phi(\frac{j\pi}{b-a})\mathcal{M}_{k,j}$ (as in (4.22)). As the integration kernel is smooth, we can approximate the integrals efficiently by, for example, a Clenshaw–Curtis quadrature rule [19]. Another way is to approximate the coefficients $\mathcal{Z}_k(t_m)$ by using a DCT.

The Euler discretization for general drift and volatility terms exhibits only first order weak convergence, which may hamper the convergence of the discretized BSDE. The usage of the simplified second order weak Taylor scheme may improve the convergence rate, and for some processes one can use an exact simulation scheme.[2]

4.5. Errors and computational complexity. In the BCOS method when solving BSDEs several approximation errors are encountered. In the first place there are discretization errors, due to the discrete-time approximation of the stochastic processes. Moreover, errors are introduced by the COS formulas and the Picard method. These error components and the computational complexity are discussed in this section.

Discretization error of the BSDE. We perform an error analysis[3] for the scheme with $\theta_1 = \theta_2 = \frac{1}{2}$ and *assume constant* μ *and* σ (see (4.20)), so that $X_m^{\Delta} = X_m$. We define the *local* theta-discretization errors in (3.5) and (3.6) by

(4.29a)
$$R_m^y(x) := \int_{t_m}^{t_{m+1}} \mathbb{E}_m^x[f(s, \mathbf{X}_s)]ds - \tfrac{1}{2}\Delta t f(t_m, \mathbf{X}_m) - \tfrac{1}{2}\Delta t \mathbb{E}_m^x[f(t_{m+1}, \mathbf{X}_{m+1})],$$

$$\begin{aligned} R_m^z(x) := &\int_{t_m}^{t_{m+1}} \mathbb{E}_m^x[f(s, \mathbf{X}_s)(\omega_s - \omega_{t_m})]ds - \tfrac{1}{2}\Delta t \mathbb{E}_m^x[f(t_{m+1}, \mathbf{X}_{m+1})\Delta\omega_{m+1}] \\ &- \int_{t_m}^{t_{m+1}} \mathbb{E}_m^x[Z_s]ds + \tfrac{1}{2}\Delta t Z_m + \tfrac{1}{2}\Delta t \mathbb{E}_m^x[Z_{m+1}]. \end{aligned} \tag{4.29b}$$

The orders of these errors depend on the smoothness of the integrands with respect to time s. If functions f and g are sufficiently smooth and bounded, with bounded deriva-

[2] This is part of forthcoming research in [42].

[3] The error analyses for other processes and other discretization schemes for the FSDE, such as the Milstein scheme, are part of forthcoming research in [42].

tives, then the absolute values of the terms $R^y_m(x)$, $R^z_m(x)$, $\frac{1}{\Delta t}\mathbb{E}^x_m[R^y_{m+1}(X_{m+1})\Delta\omega_{m+1}]$, and $(\frac{1}{\Delta t}R^z_m(x) - \mathbb{E}^x_m[R^z_{m+1}(X_{m+1})])$ can be bounded by $C(\Delta t)^3$, with C a constant depending only on T, functions g and f, and μ, σ (similar to [51, 49]).

The *global* errors due to the theta-time-discretization in (3.7c) and (3.7b) are denoted by

$$\begin{aligned}&\epsilon^y_m(X_m) := Y_m(X_m) - Y^\Delta_m(X_m), \quad \epsilon^z_m(X_m) := Z_m(X_m) - Z^\Delta_m(X_m),\\ &\epsilon^f_m(X_m) := f(t_m, \mathbf{X}_m) - f(t_m, \mathbf{X}^\Delta_m).\end{aligned} \tag{4.30}$$

We omit the dependency of the local and global errors on the state of the FSDE for notational convenience. For the y-component we have ($m \leq M-1$)

$$\epsilon^y_m = \mathbb{E}^x_m[\epsilon^y_{m+1}] + \tfrac{1}{2}\Delta t\epsilon^f_m + \tfrac{1}{2}\Delta t\mathbb{E}^x_m[\epsilon^f_{m+1}] + R^y_m. \tag{4.31}$$

With the Lipschitz assumption on driver function f, this error can be bounded, for $\frac{1}{2}\Delta tL_f < 1$, by

$$\begin{aligned}|\epsilon^y_m| \leq\ &\frac{1+\frac{1}{2}\Delta tL_f}{1-\frac{1}{2}\Delta tL_f}\mathbb{E}^x_m[|\epsilon^y_{m+1}|] + \frac{\frac{1}{2}\Delta tL_f}{1-\frac{1}{2}\Delta tL_f}|\epsilon^z_m| + \frac{\frac{1}{2}\Delta tL_f}{1-\frac{1}{2}\Delta tL_f}\mathbb{E}^x_m[|\epsilon^z_{m+1}|]\\ &+ \frac{1}{1-\frac{1}{2}\Delta tL_f}C(\Delta t)^3.\end{aligned} \tag{4.32}$$

For the z-component we have

$$\epsilon^z_m = \tfrac{2}{\Delta t}\mathbb{E}^x_m[\epsilon^y_{m+1}\Delta\omega_{m+1}] + \mathbb{E}^x_m[\epsilon^f_{m+1}\Delta\omega_{m+1}] - \mathbb{E}^x_m[\epsilon^z_{m+1}] + \tfrac{2}{\Delta t}R^z_m. \tag{4.33}$$

Substituting the similar equations for ϵ^y_{m+1} and ϵ^z_{m+1} as in (4.31) and (4.33) gives ($m \leq M-2$)

$$\begin{aligned}\epsilon^z_m =\ &\frac{2}{\Delta t}\mathbb{E}^x_m[\epsilon^y_{m+2}\Delta\omega_{m+1}] + \mathbb{E}^x_m[\epsilon^f_{m+1}\Delta\omega_{m+1}] + \mathbb{E}^x_m[\epsilon^f_{m+2}\Delta\omega_{m+1}]\\ &+ \mathbb{E}^x_m[\epsilon^f_{m+1}\Delta\omega_{m+1}] - \frac{2}{\Delta t}\mathbb{E}^x_m[\epsilon^y_{m+2}\Delta\omega_{m+2}] - \mathbb{E}^x_m[\epsilon^f_{m+2}\Delta\omega_{m+2}] + \mathbb{E}^x_m[\epsilon^z_{m+2}]\\ &+ \frac{2}{\Delta t}\mathbb{E}^x_m[R^y_{m+1}\Delta\omega_{m+1}] - \frac{2}{\Delta t}\mathbb{E}^x_m[R^z_{m+1}] + \frac{2}{\Delta t}R^z_m.\end{aligned} \tag{4.34}$$

Error ϵ^y_{m+2} is a function of X_{m+2}. The equalities (A.4) and (A.5) in Appendix A.1 then give us

$$\begin{aligned}\mathbb{E}^x_m[\epsilon^y_{m+2}\Delta\omega_{m+1}] &= \mathbb{E}^x_m[Y_{m+2}(X_{m+2})\Delta\omega_{m+1} - Y^\Delta_{m+2}(X_{m+2})\Delta\omega_{m+1}]\\ &= \sigma\Delta t\mathbb{E}^x_m[D_xY_{m+2}(X_{m+2}) - D_xY^\Delta_{m+2}(X_{m+2})]\\ &= \mathbb{E}^x_m[Y_{m+2}(X_{m+2})\Delta\omega_{m+2} - Y^\Delta_{m+2}(X_{m+2})\Delta\omega_{m+2}]\\ &= \mathbb{E}^x_m[\epsilon^y_{m+2}\Delta\omega_{m+2}].\end{aligned} \tag{4.35}$$

We can also write error ϵ^f_{m+2} as a function of X_{m+2}, as

$$\begin{aligned}\epsilon^f_{m+2} &= f(t_{m+2}, \mathbf{X}_{m+2}) - f(t_{m+2}, \mathbf{X}^\Delta_{m+2})\\ &= f(t_{m+2}, X_{m+2}, Y_{m+2}(X_{m+2}), Z_{m+2}(X_{m+2}))\\ &\quad - f(t_{m+2}, X_{m+2}, Y^\Delta_{m+2}(X_{m+2}), Z^\Delta_{m+2}(X_{m+2})).\end{aligned} \tag{4.36}$$

The equalities in Appendix A.1 result in

$$\begin{aligned}\mathbb{E}_m^x[\epsilon^f_{m+2}\Delta\omega_{m+1}] &= \sigma\Delta t\mathbb{E}_m^x\left[\frac{d}{dx}f(t_{m+2},\mathbf{X}_{m+2}) - \frac{d}{dx}f(t_{m+2},\mathbf{X}^{\Delta}_{m+2})\right]\\ &= \mathbb{E}_m^x[\epsilon^f_{m+2}\Delta\omega_{m+2}]. \qquad (4.37)\end{aligned}$$

Here $\frac{d}{dx}f$ denotes the total derivative of driver f to state x, where y and z also depend on x. With the two equalities (4.35) and (4.37) we find

(4.38)

$$\epsilon^z_m = 2\mathbb{E}_m^x[\epsilon^f_{m+1}\Delta\omega_{m+1}] + \mathbb{E}_m^x[\epsilon^z_{m+2}] + \frac{2}{\Delta t}\mathbb{E}_m^x[R^y_{m+1}\Delta\omega_{m+1}] + \frac{2}{\Delta t}\left(R^z_m - \mathbb{E}_m^x[R^z_{m+1}]\right).$$

We can bound the absolute value of the first term by

$$\begin{aligned}&|2\mathbb{E}_m^x[\epsilon^f_{m+1}\Delta\omega_{m+1}]| \le 2\mathbb{E}_m^x[|\epsilon^f_{m+1}||\Delta\omega_{m+1}|] \le 2\sup|\epsilon^f_{m+1}|\sqrt{\Delta t}\\ &\le 2L_f(\sup|\epsilon^y_{m+1}| + \sup|\epsilon^z_{m+1}|)\sqrt{\Delta t} := 2L_f(|\epsilon^y_{m+1}|_\infty + |\epsilon^z_{m+1}|_\infty)\sqrt{\Delta t}, \qquad (4.39)\end{aligned}$$

where the suprema are taken under the condition $X_m = x$.

We can now bound the absolute error by

$$|\epsilon^z_m| \le 2L_f\sqrt{\Delta t}(|\epsilon^y_{m+1}|_\infty + |\epsilon^z_{m+1}|_\infty) + \mathbb{E}_m^x[|\epsilon^z_{m+2}|] + 2C(\Delta t)^3. \qquad (4.40)$$

Next we sum up the errors. For $\frac{1}{2}\Delta t L_f < \frac{1}{2}$, $\Delta t \le 1$, there exist constants C_1 and C_2, depending on L_f, with $(m \le M-3)$

$$\begin{aligned}&\mathbb{E}_m^x[|e_m|] := \mathbb{E}_m^x[|\epsilon^y_m|_\infty + \sqrt{\Delta t}|\epsilon^z_{m+1}|_\infty + \sqrt{\Delta t}|\epsilon^z_m|_\infty]\\ &\le A\mathbb{E}_m^x[|\epsilon^y_{m+2}|_\infty + \sqrt{\Delta t}|\epsilon^z_{m+2}|_\infty + \sqrt{\Delta t}|\epsilon^z_{m+3}|_\infty] + B = A\mathbb{E}_m^x[|e_{m+2}|] + B, \qquad (4.41)\\ &\text{with } A = \frac{1}{1-L_f\Delta t}(1+C_1\Delta t),\\ &\qquad B = \frac{1}{1-L_f\Delta t}C_2(\Delta t)^3.\end{aligned}$$

THEOREM 1. *Given*

$$\mathbb{E}^x_{M-1}[|\epsilon^z_M|] \sim \mathcal{O}\left((\Delta t)^3\right), \quad \mathbb{E}^x_{M-1}[|\epsilon^y_M|] \sim \mathcal{O}\left((\Delta t)^3\right), \qquad (4.42)$$

then

$$\mathbb{E}_0^x\left[|\epsilon^y_m| + \sqrt{\Delta t}|\epsilon^z_m|\right] \le Q(\Delta t)^2, \quad 1 \le m \le M, \qquad (4.43)$$

with Q a constant only depending on T, functions g and f, and μ, σ.

Proof. With (4.31), (4.33), (4.40) it is straightforward to show that

$$\mathbb{E}^x_{M-1}[|e_{M-1}|] \sim \mathcal{O}\left((\Delta t)^2\right) \text{ and } \mathbb{E}^x_{M-2}[|e_{M-2}|] \sim \mathcal{O}\left((\Delta t)^2\right). \qquad (4.44)$$

By induction we find

(4.45)

$$\mathbb{E}_m^x[|e_m|] \le A^{\frac{1}{2}(M-m)}\mathbb{E}_m^x[|e_{M-1}| + |e_{M-2}|] + \frac{A^{\frac{1}{2}(M-m)}-1}{A-1}B \quad \text{for } m \le M-3.$$

It follows that ($1 \leq k \leq M$, $\Delta t M = T$)

$$A^k - 1 \leq A^k \leq \left(\frac{1 + C_1\Delta t}{1 - L_f\Delta t}\right)^k = \left(1 + \frac{(C_1 + L_f)\Delta t}{1 - L_f\Delta t}\right)^k$$

$$\leq \exp\left(\frac{(C_1 + L_f)\Delta t k}{1 - L_f\Delta t}\right) \leq \exp\left(\frac{(C_1 + L_f)T}{1 - \Delta t L_f}\right) \tag{4.46}$$

is bounded and

$$\frac{B}{A - 1} \leq \frac{C_2}{C_1 + L_f}(\Delta t)^2. \qquad \square \tag{4.47}$$

The authors in [49] obtain second order convergence in both Y and Z terms for the case that the FSDE equals the Wiener process. Convergence of (Y^Δ, Z^Δ) to (Y, Z) is discussed in [8, 48, 29, 21, 7] for the special case $\theta_1 = \theta_2 = 1$. Under certain conditions on functions f and g, error convergence of order $\mathcal{O}\left((\Delta t)^{1/2}\right)$ in the L^2-sense was found. The authors in [10] prove convergence of a discrete scheme with a scaled random walk using a Donsker-type theorem. For the error analysis of other schemes and L^p-errors, we refer the reader to [47, 31].

Error in COS formulas. In section 3 we encountered deterministic functions y and z, such that

$$y(t_m, X_m^\Delta) = Y_m^\Delta(X_m^\Delta), \quad z(t_m, X_m^\Delta) = Z_m^\Delta(X_m^\Delta).$$

These functions are approximated by COS formulas, and the corresponding Fourier coefficients are recovered backward in time, resulting in the approximations

$$\hat{y}(t_m, X_m^\Delta) \quad \text{and} \quad \hat{z}(t_m, X_m^\Delta). \tag{4.48}$$

The errors of these numerical approximations are denoted by

$$\epsilon^y_{COS}(t_m, X_m^\Delta) := y(t_m, X_m^\Delta) - \hat{y}(t_m, X_m^\Delta), \tag{4.49a}$$

$$\epsilon^z_{COS}(t_m, X_m^\Delta) := z(t_m, X_m^\Delta) - \hat{z}(t_m, X_m^\Delta). \tag{4.49b}$$

Fourier series expansions and their convergence properties have been discussed in [9]. Errors of the COS method are introduced in three steps (see section 4.1): the truncation of the integration range, the substitution of the density by its cosine series expansion on the truncated range, and the substitution of the series coefficients by the characteristic function approximation. A detailed error analysis was given in [16, 17] and in the supplementary material [43]. For a sufficiently wide computational domain $[a, b]$ the truncation error in our domain of interest can be neglected, because the truncated mass of the density function is negligible. The error component $I - \hat{I}$ (equations (4.10)) converges exponentially in the number of terms in the series expansions for smooth density functions and a sufficiently wide integration interval. The transitional density that is related to the Euler scheme is smooth and results in exponential convergence in N. A density function with a discontinuity in one of its derivatives gives rise to an algebraic convergence in N. We refer the reader to [44] for more information on the convergence of discontinuous functions. Algorithm 1 explains how to recover the coefficients $\mathcal{Z}_k(t_m)$, $\mathcal{Y}_k(t_m)$, and $\mathcal{F}_k(t_m)$ backward in time. This introduces an additional error. The use of DCTs to approximate the Fourier cosine coefficients gives an error with algebraic index of convergence two in N, as we demonstrate by an example in [43] (section 3).

Convergence of Picard iterations. With P Picard iterations we find the fixed point y of the equation

$$y = \Delta t \theta_1 f(t_m, x, y, z(t_m, x)) + h(t_m, x). \tag{4.50}$$

The driver function f is assumed to be Lipschitz in y and z, with Lipschitz constant L_f. For Δt small enough, i.e., $L_f \Delta t \theta_1 < 1$, a unique fixed point exists, and the Picard iterations converge toward that point for any initial guess. The fixed-point technique converges to the true solution at the geometric rate $\Delta t \theta_1 L_f$, which depends on the Lipschitz condition of the driver function.

Total error. The absolute value of the total errors can be bounded by

(4.51a)
$$|\varepsilon^y_m(X_m, X^\Delta_m)| := |Y_m(X_m) - \hat{y}(t_m, X^\Delta_m)| \leq |Y_m(X_m) - Y^\Delta_m(X^\Delta_m)| + |\epsilon^y_{COS}(t_m, X^\Delta_m)|,$$

(4.51b)
$$|\varepsilon^z_m(X_m, X^\Delta_m)| := |Z_m(X_m) - \hat{z}(t_m, X^\Delta_m)| \leq |Z_m(X_m) - Z^\Delta_m(X^\Delta_m)| + |\epsilon^z_{COS}(t_m, X^\Delta_m)|.$$

For the numerical experiments in section 5 we take N sufficiently high. Then we can neglect the errors ϵ_{COS} and are able to investigate the error of the discretization scheme.

Computational complexity. The computation time of the BCOS method is linear in the number of timesteps M. For each discrete time-point t_m we perform the following operations:

- Computation of $\hat{z}(t_m, x)$ and $\hat{h}(t_m, x)$ on an x-grid, in $\mathcal{O}(N^2)$ operations.
- Initialization Picard method: Computation of $\hat{y}^0(t_m, x)$ on an x-grid, in $\mathcal{O}(N^2)$ operations.
- Computation of $\hat{y}^P(t_m, x)$ on an x-grid by P Picard iterations, in $\mathcal{O}(PN)$ operations.
- Computation of $\mathcal{Z}_k(t_m)$ and $\mathcal{H}_k(t_m)$ by the FFT algorithm, in $\mathcal{O}(N \log N)$ operations.
- Computation of $\mathcal{F}_k(t_m) \approx \mathcal{F}^{P-1}_k(t_m)$ by DCT (see [43]), in $\mathcal{O}(N \log N)$ operations.
- Computation of $\mathcal{Y}_k(t_m) \approx \mathcal{Y}^P_k(t_m)$, in $\mathcal{O}(N)$ operations.

For the approximation of the coefficients $\mathcal{F}^{P-1}_k(t_m)$ in (4.26) by a DCT we first need to compute $\hat{z}(t_m, x)$, $\hat{h}(t_m, x)$, and $\hat{y}^0(t_m, x)$ on an x-grid with N equidistant points, which is of order $\mathcal{O}(N^2)$. This is the most time-consuming part of the algorithm. However, these functions can be computed in parallel. In total, the complexity of the BCOS method, Algorithm 1, is $\mathcal{O}\left(M\left(N^2 + PN + N\log N + N\log N + N\right)\right)$.

5. Numerical experiments.

In this section we discuss two numerical experiments. MATLAB 7.11.0 is used for the computations, with an Intel(R) Core(TM) i5-2520M CPU @ 2.50GHz and 7.7 GB RAM. To test the general theta-method we distinguish between four discretization schemes:

Scheme A:	$\theta_1 = 0,$	$\theta_2 = 1,$	Scheme C:	$\theta_1 = 1,$	$\theta_2 = 1,$
Scheme B:	$\theta_1 = 0.5,$	$\theta_2 = 1,$	Scheme D:	$\theta_1 = 0.5,$	$\theta_2 = 0.5.$

For all four schemes, $z(t_m, x)$ can be solved explicitly, and $y(t_m, x)$ is solved explicitly for scheme A and implicitly with $P = 5$ Picard iterations for the other schemes.

Similarly to [16], we prescribe a computational domain $[a, b]$ by

$$[a, b] = [x_0 + \kappa_1 - L\sqrt{\kappa_2},\, x_0 + \kappa_1 + L\sqrt{\kappa_2}], \tag{5.1}$$

with cumulants $\kappa_1 = \mu T$ and $\kappa_2 = \sigma^2 T$, and $L = 10$. Furthermore, we set the number of terms in the Fourier cosine series expansions equal to $N = 2^9$. For these values the BCOS method has converged in N to machine precision.

5.1. Example 1. The first example is taken from [50]. The underlying process is the Wiener process, i.e., $X_t = \omega_t$. The BSDE reads

$$dY_t = -f(t, X_t, Y_t, Z_t)dt + Z_t d\omega_t, \tag{5.2a}$$

$$f(t, X_t, Y_t, Z_t) = Y_t Z_t - Z_t + 2.5Y_t - \sin(t + X_t)\cos(t + X_t) - 2\sin(t + X_t), \tag{5.2b}$$

$$Y_T = g(X_T) = \sin(X_T + T). \tag{5.2c}$$

The exact solution is given by

$$(Y_t,\, Z_t) = (\sin(X_t + t),\, \cos(X_t + t)). \tag{5.3}$$

We take terminal time $T = 1$, which gives $(Y_0,\, Z_0) = (0,\, 1)$. Note that driver $f(.)$ depends also on time t and state X_t. For the results of the BCOS method, we refer to Figure 1. We observe that the approximated value $\hat{y}(t_0, x_0)$ converges with $\mathcal{O}(\Delta t)$ for the schemes A, B, and C and $\mathcal{O}((\Delta t)^2)$ for scheme D. The approximated value $\hat{z}(t_0, x_0)$ converges with $\mathcal{O}((\Delta t)^2)$ for scheme D and with $\mathcal{O}(\Delta t)$ for the other three schemes, which is in accordance with the error analysis in section 4.5.

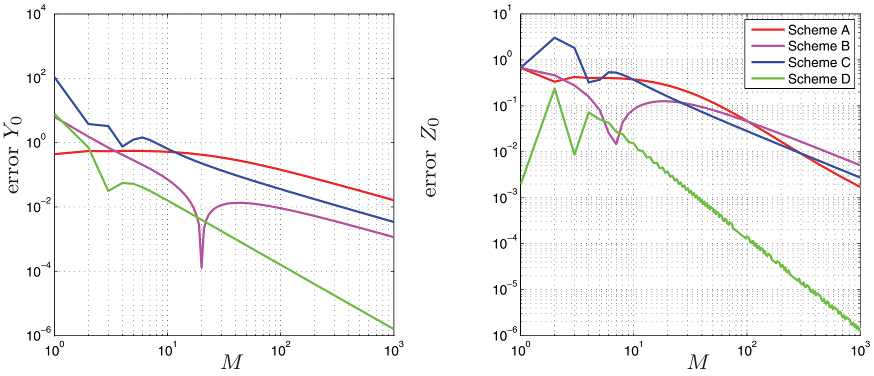

Fig. 1. *Results for example 1 ($N = 2^9$). Left: Error in $\hat{y}(t_0, x_0)$. Right: Error in $\hat{z}(t_0, x_0)$. (Color available in electronic version.)*

Table 1 shows CPU times for scheme D, for different values of M and N. Each test required less than one second. Computation of the functions $\hat{z}(t_m, x)$, $\hat{h}(t_m, x)$, and $\hat{y}^0(t_m, x)$ on an x-grid is the most time-consuming part of the algorithm. The computation time is linear in the number of time steps M and of $\mathcal{O}(N \log N)$ order in the number of terms in the Fourier cosine series expansions.

5.2. Example 2: Black–Scholes call option. In this example we compute the price $v(t, S_t)$ of a call option by a BSDE where the underlying asset follows a geometric Brownian motion,

$$dS_t = \bar{\mu} S_t dt + \bar{\sigma} S_t d\omega_t. \tag{5.4}$$

The exact solution is given by the Black–Scholes price, which is known analytically [6]. For the derivation of the Black–Scholes PDE we set up a self-financing portfolio

TABLE 1
CPU time (s).

M	4	8	16	32	64	128	256	512
$N = 2^9$	0.0301	0.0304	0.0412	0.0639	0.1071	0.1966	0.3736	0.7292

N	2^6	2^7	2^8	2^9
$M = 256$	0.0940	0.1109	0.1552	0.3736

Y_t with a_t assets and bonds with risk-free return rate r. Markets are assumed to be complete in this model, there are no trading restrictions, and the option can be exactly replicated by the hedging portfolio, that is, $Y_T = \max(S_T - K, 0)$. Then, the option value at the initial time should be equal to the initial value of the portfolio. The portfolio evolves according to the stochastic differential equation

$$dY_t = r(Y_t - a_t S_t)dt + a_t dS_t = \left(rY_t + \frac{\bar{\mu} - r}{\bar{\sigma}}\bar{\sigma} a_t S_t\right)dt + \bar{\sigma} a_t S_t d\omega_t. \tag{5.5}$$

If we set $Z_t = \bar{\sigma} a_t S_t$, then (Y, Z) solves the BSDE

$$dY_t = -f(t, S_t, Y_t, Z_t)dt + Z_t d\omega_t, \tag{5.6a}$$

$$f(t, S_t, Y_t, Z_t) = -rY_t - \frac{\bar{\mu} - r}{\bar{\sigma}} Z_t, \tag{5.6b}$$

$$Y_T = \max(S_T - K, 0). \tag{5.6c}$$

Y_t corresponds to the value of the portfolio, and Z_t is related to the hedging strategy. In this case, the driver function $f(.)$ is Lipschitz continuous and linear with respect to y and z. The option value is given by $v(t, S_t) = Y_t$ and $\bar{\sigma} S_t v_S(t, S_t) = Z_t$. For the tests, we use the parameter values

$$S_0 = 100,\ K = 100,\ r = 0.1,\ \bar{\mu} = 0.2,\ \bar{\sigma} = 0.25,\ T = 0.1, \tag{5.7}$$

with the exact solutions $Y_0 = v(t_0, S_0) = 3.65997$ and $Z_0 = \bar{\sigma} S_0 v_S(t_0, S_0) = 14.14823$. For the numerical approximation, we switch to the log-asset domain $X_t = \log S_t$, with

$$dX_t = (\bar{\mu} - \tfrac{1}{2}\bar{\sigma}^2)dt + \bar{\sigma} d\omega_t. \tag{5.8}$$

Results of the BCOS method for all four schemes are presented in Figure 2. The approximated values $\hat{y}(t_0, x_0)$ and $\hat{z}(t_0, x_0)$ both converge with $\mathcal{O}(\Delta t)$ for schemes A, B, and C and with $\mathcal{O}((\Delta t)^2)$ for scheme D, as expected.

We would like to emphasize that solving the BSDE is done under the historical, real-world $\mathbb{P}$-measure. However, the exact Black–Scholes solution does not depend on $\bar{\mu}$. In Figure 3 we see results for different values of drift $\bar{\mu}$. The convergence rates in M are the same, but a higher value of $\bar{\mu}$ gives a larger error for the same number of time steps M. This is due to the Lipschitz constant $L_f = \max(\frac{\bar{\mu} - r}{\bar{\sigma}}, r)$, which is increasing in $\bar{\mu}$.

6. Exponential utility maximization and utility indifference price. In a financial market with jumps or with constrained hedging strategies it is usually not possible to perform a perfect hedge which exactly attains the option payoff as the final value; there is a so-called replication error. If markets are not complete, there are different ways to value options [13]:

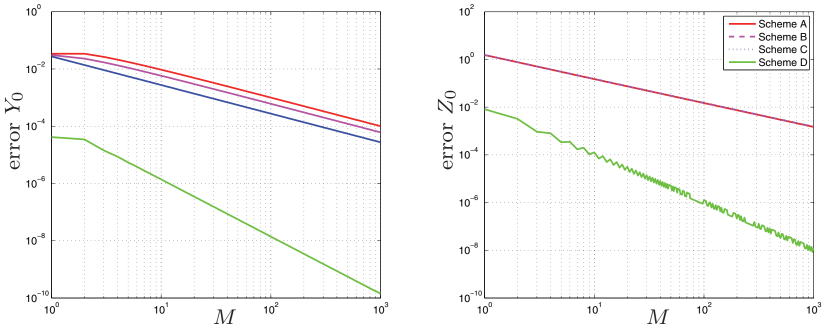

FIG. 2. *Results for example 2 ($N = 2^9$). Left: Error in $\hat{y}(t_0, x_0)$. Right: Error in $\hat{z}(t_0, x_0)$. (Color available in electronic version.)*

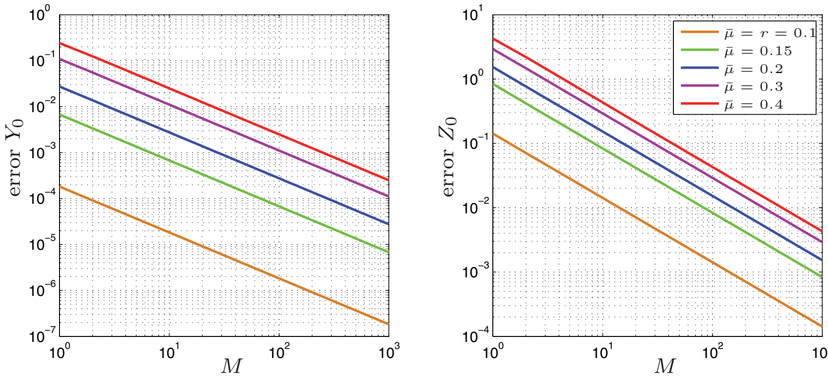

FIG. 3. *Results for example 2 for different values of $\bar{\mu}$ (scheme C). (Color available in electronic version.)*

- *Superstrategies* are strategies with a positive replication error. The super-replicating option price is the minimal initial investment to find a strategy that always dominates the payoff of the option [14].
- *Risk-minimizing strategies* are used when the problem requires a strategy with minimal variance for the replication error. They were first introduced by Föllmer and Sondermann in [18].
- *Utility indifference pricing* maximizes the utility of the replication error. The corresponding price makes an agent indifferent in terms of expected utility between selling the option or not selling it. Utility indifference pricing was introduced by Hodges and Neuberger in [24].

We focus on utility indifference pricing, which basically consists of solving two utility maximization problems, one with and one without an option liability. In the next section we consider a general utility maximization problem. We employ the model of Morlais in [34], making use of an exponential utility function and jumps in the asset price. The problem can be defined by a BSDE including jumps. We refer the reader to [39, 45, 25, 33, 38] for the setting where asset prices follow only a diffusion process. This model is generalized by jumps in [2, 34].

6.1. Exponential utility maximization under jump-diffusion with option payoff. Following the notation in [34], the probability space $(\Omega, \mathbb{F}, \mathbb{P})$ is now equipped with two independent stochastic processes: the standard Brownian motion ω and a real-valued Poisson point process defined on $\Omega \times [0,T] \times E$, with $E := \mathbb{R}/\{0\}$. We denote by $N(dJ, dt)$ the associated Poisson random measure whose compensator is assumed to be of the form $\nu(dJ)dt$, where $\nu(dJ)$ stands for the Lévy measure, which is positive and satisfies

$$\nu(\{0\}) = 0 \quad \text{and} \quad \int_E (1 \wedge |J|)^2 \nu(dJ) < \infty. \tag{6.1}$$

$N(B,t)$, $B \subset \mathbb{R}$, is the number of jumps with size in set B which occur before or at time t, and $\nu(B)$ counts the expected number of jumps in a unit time interval. $\mathcal{F}$ is the completed filtration generated by both processes ω and N. The so-called compensated Poisson random measure, $\tilde{N}$, is given by

$$\tilde{N}(dJ, dt) = N(dJ, dt) - \nu(dJ)dt. \tag{6.2}$$

The asset price is supposed to follow the jump-diffusion process

$$dS_t/S_{t-} = b(t)dt + \sigma(t)d\omega_t + \int_E \beta(t,J)\tilde{N}(dJ,dt); \tag{6.3}$$

S_{t-} represents the value of the asset just before a possible jump occurs. The jumps may model the occurrence of, for example, market crashes or default losses. An agent sells a bounded $\mathcal{F}_T$-measurable option payoff $\xi = g(S_T)$ at time $t = 0$. He is endowed with some initial capital w and then invests α_t, $t \in [0,T]$, of his portfolio W_t^α in assets, where the superscript emphasized the dependence on α. The remaining part is invested in a risk-free opportunity with zero rate of return, i.e., $r = 0$. The dynamics of this self-financing portfolio read

$$dW_t^\alpha = \alpha_t \frac{dS_t}{S_{t-}} = \alpha_t b(t)dt + \alpha_t \sigma(t) d\omega_t + \alpha_t \int_E \beta(t,J)\tilde{N}(dJ,dt), \quad W_0^\alpha = w. \tag{6.4}$$

At terminal time T there is an uncertain claim ξ, and the agent is able to reduce the risk by his trading strategy. The attitude of the agent toward possible profits and losses is measured by an exponential utility,

$$\mathfrak{U}(x) = -\exp(-\eta x), \quad \eta > 0. \tag{6.5}$$

The utility function is monotonically increasing and concave; η is the coefficient of absolute risk aversion and represents the degree of risk aversion. A higher value of η corresponds to a higher level of risk aversion. A negative amount of final wealth has a higher weight than a positive amount; in other words, more weight is given to unfavorable losses. $\eta = 0$ corresponds to risk neutrality and $\eta = \infty$ to absolute risk aversion. The agent wants to maximize his expected utility at time T, and his *objective function* now reads

$$V(w) = \max_{\alpha \in \mathcal{A}} \mathbb{E}\left[\mathfrak{U}(W_T^\alpha - \xi)\right] = \max_{\alpha \in \mathcal{A}} \mathbb{E}\left[\mathfrak{U}\left(w + \int_0^T \alpha_t \frac{dS_t}{S_{t-}} - \xi\right)\right], \tag{6.6}$$

where we maximize over the investment opportunities α in the constraint set $\mathcal{A}$ with admissible strategies. Possible trading strategies may be restricted; for example, an agent may be forced not to hold a negative number of assets. For the tests in section 7 we will take $\mathcal{A} = [\alpha_{min},\, \alpha_{max}]$.

The objective function can also be characterized by a backward stochastic differential equation with jumps (BSDEJ), as follows:

$$V(w) = \mathfrak{U}(w - Y_0) = -e^{-\eta w} e^{\eta Y_0}, \tag{6.7}$$

where (Y, Z, U) is the solution to a BSDEJ, given by [34]

$$dY_t = -f(t, Z_t, U_t)dt + Z_t d\omega_t + \int_E U_t(J)\tilde{N}(dJ, dt), \quad Y_T = \xi, \tag{6.8a}$$

$$\begin{aligned} f(t, z, u) = &-z\frac{b(t)}{\sigma(t)} - \frac{1}{2\eta}\left|\frac{b(t)}{\sigma(t)}\right|^2 \\ &+ \min_{\alpha \in \mathcal{A}} \left[\frac{\eta}{2}\left(\alpha\sigma(t) - \left(z + \frac{1}{\eta}\frac{b(t)}{\sigma(t)}\right)\right)^2 + |u(.) - \alpha\beta(t, .)|_\eta\right], \end{aligned} \tag{6.8b}$$

$$|u(.)|_\eta = \int_E \frac{\exp(\eta u(J)) - \eta u(J) - 1}{\eta}\nu(dJ). \tag{6.8c}$$

The solution of the above BSDEJ consists of a triplet of processes (Y, Z, U) in $\mathcal{S}^\infty(\mathbb{R}) \times L^2(\omega) \times L^2(\tilde{N})$.[4] Existence and uniqueness results for this BSDEJ are provided in [34]. For more information about existence and uniqueness of BSDEJs, we refer the reader to [46, 1, 40]. Furthermore, there exists an optimal predictable strategy $\alpha_t^* \in \mathcal{A}$ that attains the minimum in (6.8c) for $(t, z, u) = (t, Z_t, U_t)$.

6.2. Utility indifference price. Now we start with the utility indifference price, where the idea is the following. The seller of an option receives the option premium and hedges the option with an optimal strategy that maximizes the utility of the portfolio value at the terminal time minus the payoff. We also determine the expected utility without the option trade. The *utility indifference price* of the option is defined as the additional initial wealth with which the seller can achieve the same utility as without the option.

Let $u_0(w)$ denote the utility maximization value without the option payoff,

$$u_0(w) = \max_{\alpha_t \in \mathcal{A}} \mathbb{E}\left[\mathfrak{U}\left(W_T^\alpha\right)\right], \tag{6.9}$$

and let $u_\xi(w)$ denote the utility maximization value in presence of the option,

$$u_\xi(w) = \max_{\alpha_t \in \mathcal{A}} \mathbb{E}\left[\mathfrak{U}\left(W_T^\alpha - \xi\right)\right]. \tag{6.10}$$

The *seller's indifference price (ask price)* v^a satisfies

$$u_0(w) = u_\xi(w + v^a). \tag{6.11}$$

In other words, it is the price at which a seller is indifferent, in the sense that the expected utility under optimal trading remains the same, between selling the option

[4] Following [34], $\mathcal{S}^\infty(\mathbb{R})$ is the set of all adapted processes Y with càdlàg paths such that $\sup_\Omega(\sup_{t\in[0,T]} |Y_t|) < \infty$. $L^2(\omega)$ is the set of all predictable processes Z such that $\mathbb{E}[\int_0^T |Z_s|^2 ds] < \infty$ $L^2(\tilde{N})$ is the set of all $\mathcal{P}\otimes\mathcal{B}(E)$-measurable processes U such that $\mathbb{E}[\int_0^T \int_E |U_s(J)|^2 \nu(dJ)ds] < \infty$. $\mathcal{P}$ stands for the σ-field of all predictable sets of $[0, T] \times \Omega$ and $\mathcal{B}(E)$ the Borel field of E.

for price v^a and not selling any option. We need to solve for v^a, and with the theory in section 6.1 we find

$$\mathfrak{U}(w - Y_0^0) = \mathfrak{U}(w + v^a - Y_0^\xi) \quad \Longrightarrow \quad v^a = Y_0^\xi - Y_0^0, \tag{6.12}$$

where Y_t^0 and Y_t^ξ follow BSDEJ (6.8) with terminal conditions $Y_T = 0$ and $Y_T = \xi$, respectively. With this we can value an option under jump-diffusion and when the trading strategies are constrained, for example, $\mathcal{A} = [\alpha_{min}, \alpha_{max}]$, with the help of BSDEJs.

The *buyer's indifference price (bid price)* v^b is defined in a similar way and satisfies

$$u_0(w) = u_{-\xi}(w - v^b). \tag{6.13}$$

Again, with the BSDE approach we find that

$$\mathfrak{U}(w - Y_0^0) = \mathfrak{U}(w - v^b - Y_0^{-\xi}) \quad \Longrightarrow \quad v^b = Y_0^0 - Y_0^{-\xi}. \tag{6.14}$$

Below is a list of properties of utility indifference prices (see, for example, [23, 39]). We here denote by $v(\eta, \xi)$ the option price with coefficient of absolute risk aversion η and option ξ.

- Prices v^b and v^a are independent of initial wealth w.
- Bid and ask prices are related via $v^b(\eta, \xi) = -v^a(\eta, -\xi)$.
- The ask price is larger than the bid price: $v^a \geq v^b$.
- If the market is complete, i.e., there are no jumps and $\mathcal{A} = \mathbb{R}$, then the option is perfectly replicable. The driver function reduces to $f(t, z, u) = -z\frac{b}{\sigma} - \frac{1}{2\eta}|\frac{b}{\sigma}|^2$, and the utility indifference prices reduce to the Black–Scholes prices.

6.3. Discretization and BCOS method for BSDEJs. In this section, we explain the BCOS method to solve BSDEJ (6.8). We suppose that the asset price follows the following FSDE:

$$dS_t/S_{t-} = bdt + \sigma d\omega_t + \int_E \beta(J)\tilde{N}(dJ, dt), \quad \text{with } \beta(J) = e^J - 1. \tag{6.15}$$

Moreover, E is assumed to be a finite set, $E = \{j_1, j_2, \dots, j_{n_j}\}$, with Lévy measure $\nu(\{j_\ell\}) = \lambda p_\ell$, where $\lambda = \nu(\mathbb{R})$ is the intensity rate. In other words, p_ℓ is the probability of jump size j_ℓ and $\nu(dJ) = \lambda \sum_{\ell=1}^{n_j} p_\ell \delta_{j_\ell}(dJ)$. So,

$$\int_E \beta(J)\tilde{N}(dJ, dt) = \sum_{\ell=1}^{n_j} \beta(j_\ell)\tilde{N}(\{j_\ell\}, dt). \tag{6.16}$$

We define $\mu := b - \int_E \beta(J)\nu(dJ)$ and switch to the log-asset domain $X_t = \log S_t$, i.e.,

$$dX_t = \left(\mu - \tfrac{1}{2}\sigma^2 + \int_E J\nu(dJ)\right) dt + \sigma d\omega_t + \int_E J\tilde{N}(dJ, dt). \tag{6.17}$$

The Euler discretization of FSDE (6.17) reads

$$X_{m+1}^\Delta = X_m^\Delta + \left(\mu - \tfrac{1}{2}\sigma^2 + \int_E J\nu(dJ)\right) \Delta t + \sigma \Delta\omega_{m+1} + \int_E J\tilde{N}(dJ, \Delta t), \tag{6.18}$$

where we defined $\tilde{N}(dJ,\Delta t) := \tilde{N}(dJ,(t_m,t_{m+1}]) = \tilde{N}(dJ,t_{m+1}) - \tilde{N}(dJ,t_m)$. The characteristic function of X^{Δ}_{m+1}, given $X^{\Delta}_m = x$, reads

$$\varphi(u|x) = \varphi(u|0)e^{iux} = \phi(u)e^{iux}, \quad \text{with}$$

$$\phi(u) := \exp\left(iu(\mu - \tfrac{1}{2}\sigma^2)\Delta t - \tfrac{1}{2}u^2\sigma^2\Delta t\right) e^{\lambda\Delta t(\varphi_J(u)-1)}, \quad \varphi_J(u) = \sum_{\ell=1}^{n_j} p_\ell e^{iuj_\ell}. \tag{6.19}$$

For the discretization of the BSDEJ, we start from

$$Y_m = Y_{m+1} + \int_{t_m}^{t_{m+1}} f(s,Z_s,U_s)ds - \int_{t_m}^{t_{m+1}} Z_s d\omega_s - \int_{t_m}^{t_{m+1}} \int_E U_s(J)\tilde{N}(dJ,ds). \tag{6.20}$$

Both processes ω and $\tilde{N}$ are independent. Taking conditional expectations of both sides of (6.20) and applying the theta-method results, similar to (3.5), in

$$\begin{aligned} Y_m \approx{}& \mathbb{E}_m[Y_{m+1}] + \Delta t\theta_1 f(t_m,Z_m,U_m) \\ &+ \Delta t(1-\theta_1)\mathbb{E}_m[f(t_{m+1},Z_{m+1},U_{m+1})],\ \theta_1 \in [0,1]. \end{aligned} \tag{6.21}$$

Multiplying both sides of (6.20) by $\Delta\omega_{m+1}$ and taking conditional expectations gives us, similar to (3.6),

$$\begin{aligned} 0 \approx{}& \mathbb{E}_m[Y_{m+1}\Delta\omega_{m+1}] + \Delta t(1-\theta_2)\mathbb{E}_m[f(t_{m+1},Z_{m+1},U_{m+1})\Delta\omega_{m+1}] \\ &- \Delta t\theta_2 Z_m - \Delta t(1-\theta_2)\mathbb{E}_m[Z_{m+1}], \quad \theta_2 \in [0,1]. \end{aligned} \tag{6.22}$$

Multiplying both sides of (6.20) by $\tilde{N}(\{j_\ell\},\Delta t)$ and taking conditional expectations gives

$$\begin{aligned} 0 ={}& \mathbb{E}_m[Y_{m+1}\tilde{N}(\{j_\ell\},\Delta t)] + \int_{t_m}^{t_{m+1}} \mathbb{E}_m\left[f(s,Z_s,U_s)\tilde{N}(\{j_\ell\},s-t_m)\right] ds \\ &- \int_{t_m}^{t_{m+1}} p_\ell\lambda\mathbb{E}_m\left[U_s(j_\ell)\right] ds, \end{aligned} \tag{6.23}$$

where we used the Itô isometry for

$$\begin{aligned} &\mathbb{E}_m\left[\int_{t_m}^{t_{m+1}}\int_E U_s(J)\tilde{N}(dJ,ds)\tilde{N}(\{j_\ell\},\Delta t)\right] \\ &= \mathbb{E}_m\left[\int_{t_m}^{t_{m+1}}\int_E U_s(J)\tilde{N}(dJ,ds)\int_{t_m}^{t_{m+1}}\int_E \delta_{j_\ell}(J)\tilde{N}(dJ,ds)\right] \\ &= \mathbb{E}_m\left[\int_{t_m}^{t_{m+1}} p_\ell\lambda U_s(j_\ell)ds\right]. \end{aligned} \tag{6.24}$$

By the theta-discretization we get

$$\begin{aligned} 0 \approx{}& \mathbb{E}_m[Y_{m+1}\tilde{N}(\{j_\ell\},\Delta t)] + \Delta t(1-\theta_3)\mathbb{E}_m[f(t_{m+1},Z_{m+1},U_{m+1})\tilde{N}(\{j_\ell\},\Delta t)] \\ &- p_\ell\lambda\Delta t\theta_3 U_m(j_\ell) - p_\ell\lambda\Delta t(1-\theta_3)\mathbb{E}_m[U_{m+1}(j_\ell)], \quad \theta_3 \in [0,1], \ \text{for } \ell = 1,\dots,n_j. \end{aligned} \tag{6.25}$$

The above equations lead to a time-discretization $(Y^\Delta, Z^\Delta, U^\Delta)$ for (Y, Z, U), as follows:

(6.26a)
$$Y_M^\Delta = g(X_M^\Delta),$$
$$\text{for } m = M-1, \dots, 0:$$
$$Z_m^\Delta = -\theta_2^{-1}(1-\theta_2)\mathbb{E}_m[Z_{m+1}^\Delta] + \tfrac{1}{\Delta t}\theta_2^{-1}\mathbb{E}_m[Y_{m+1}^\Delta \Delta\omega_{m+1}]$$
(6.26b)
$$+ \theta_2^{-1}(1-\theta_2)\mathbb{E}_m[f(t_{m+1}, Z_{m+1}^\Delta, U_{m+1}^\Delta)\Delta\omega_{m+1}],$$
$$U_m^\Delta(j_\ell) = -\theta_3^{-1}(1-\theta_3)\mathbb{E}_m[U_{m+1}^\Delta(j_\ell)] + \tfrac{1}{p_\ell\lambda\Delta t}\theta_3^{-1}\mathbb{E}_m[Y_{m+1}^\Delta \tilde{N}(\{j_\ell\}, \Delta t)]$$
(6.26c)
$$+ \tfrac{1}{p_\ell\lambda}\theta_3^{-1}(1-\theta_3)\mathbb{E}_m[f(t_{m+1}, Z_{m+1}^\Delta, U_{m+1}^\Delta)\tilde{N}(\{j_\ell\}, \Delta t)], \quad \ell = 1, \dots, n_j,$$
(6.26d)
$$Y_m^\Delta = \mathbb{E}_m[Y_{m+1}^\Delta] + \Delta t\theta_1 f(t_m, Z_m^\Delta, U_m^\Delta) + \Delta t(1-\theta_1)\mathbb{E}_m[f(t_{m+1}, Z_{m+1}^\Delta, U_{m+1}^\Delta)].$$

As the terminal condition is a deterministic function of X_M^Δ and because X^Δ is a Markov process, it is easily seen that there are deterministic functions $y(t_m, x)$, $z(t_m, x)$, and $u(t_m, x, j_\ell)$ so that

(6.27)
$$Y_m^\Delta = y(t_m, X_m^\Delta), \quad Z_m^\Delta = z(t_m, X_m^\Delta), \quad U_m^\Delta(j_\ell) = u(t_m, X_m^\Delta, j_\ell), \quad \ell = 1, \dots, n_j.$$

So, the random variables Y_m^Δ, Z_m^Δ, and $U_m^\Delta(j_\ell)$ are functions of X_m^Δ for each $m = 0, \dots, M$. The functions $y(t_m, x)$, $z(t_m, x)$, and $u(t_m, x, j_\ell)$ are obtained in a backward manner. Similar to section 4, the Fourier cosine coefficients of the functions $z(t_m, x)$, $f(t_m, z(t_m, x), u(t_m, x, .))$, and $y(t_m, x)$ are denoted by $\mathcal{Z}_k(t_m)$, $\mathcal{F}_k(t_m)$, and $\mathcal{Y}_k(t_m)$, respectively. Let $\mathcal{U}_k^\ell(t_m)$ be the Fourier cosine coefficients of $u(t_m, x, j_\ell)$, i.e.,

$$\mathcal{U}_k^\ell(t_m) = \frac{2}{b-a}\int_a^b u(t_m, x, j_\ell)\cos\left(k\pi\frac{x-a}{b-a}\right)dx, \quad \ell = 1, \dots, n_j. \tag{6.28}$$

We obtain the following COS formulas to approximate the conditional expectations in (6.26c); see Appendix A.2 for details.

(6.29a)
$$\mathbb{E}_m^x[U_{m+1}^\Delta(j_\ell)] \approx {\sum_{k=0}^{N-1}}' \mathcal{U}_k^\ell(t_{m+1})\Re\left(\phi\left(\frac{k\pi}{b-a}\right)e^{ik\pi\frac{x-a}{b-a}}\right),$$
(6.29b)
$$\mathbb{E}_m^x\left[Y_{m+1}^\Delta\tilde{N}(\{j_\ell\}, \Delta t)\right] \approx {\sum_{k=0}^{N-1}}' \mathcal{Y}_k(t_{m+1})\Re\left\{\phi\left(\frac{k\pi}{b-a}\right)e^{ik\pi\frac{x-a}{b-a}}p_\ell\lambda\Delta t\left[\exp\left(i\frac{k\pi j_\ell}{b-a}\right)-1\right]\right\},$$
$$\mathbb{E}_m^x\left[f(t_{m+1}, Z_{m+1}^\Delta, U_{m+1}^\Delta)\tilde{N}(\{j_\ell\}, \Delta t)\right]$$
(6.29c)
$$\approx {\sum_{k=0}^{N-1}}' \mathcal{F}_k(t_{m+1})\Re\left\{\phi\left(\frac{k\pi}{b-a}\right)e^{ik\pi\frac{x-a}{b-a}}p_\ell\lambda\Delta t\left[\exp\left(i\frac{k\pi j_\ell}{b-a}\right)-1\right]\right\}.$$

Furthermore, we use the COS formulas (4.15) and (4.17) from sections 4.2 and 4.3, obtained with the equality in Appendix A.1, to find

$$\begin{aligned}z(t_m,x) \approx{}& -\frac{1-\theta_2}{\theta_2}\frac{b-a}{2}\sum_{k=0}^{N-1}{}' \mathcal{Z}_k(t_{m+1})\Phi_k(x) + \frac{1}{\Delta t\theta_2}\frac{b-a}{2}\sum_{k=0}^{N-1}{}' \mathcal{Y}_k(t_{m+1})\sigma\Delta t\Phi_k'(x)\\ &+ \frac{1-\theta_2}{\theta_2}\frac{b-a}{2}\sum_{k=0}^{N-1}{}' \mathcal{F}_k(t_{m+1})\sigma\Delta t\Phi_k'(x),\end{aligned} \tag{6.30a}$$

$$\begin{aligned}u(t_m,x,j_\ell) \approx{}& -\frac{1-\theta_3}{\theta_3}\frac{b-a}{2}\sum_{k=0}^{N-1}{}' \mathcal{U}_k^\ell(t_{m+1})\Phi_k(x)\\ &+ \sum_{k=0}^{N-1}{}' \left(\frac{1}{\Delta t\theta_3}\mathcal{Y}_k(t_{m+1}) + \frac{1-\theta_3}{\theta_3}\mathcal{F}_k(t_{m+1})\right)\\ &\cdot \Re\left\{\phi\left(\frac{k\pi}{b-a}\right)e^{ik\pi\frac{x-a}{b-a}}\Delta t\left[\exp\left(i\frac{k\pi j_\ell}{b-a}\right)-1\right]\right\},\end{aligned} \tag{6.30b}$$

$$\begin{aligned}y(t_m,x) \approx{}& \frac{b-a}{2}\sum_{k=0}^{N-1}{}' \left(\mathcal{Y}_k(t_{m+1}) + \Delta t(1-\theta_1)\mathcal{F}_k(t_{m+1})\right)\Phi_k(x)\\ &+ \Delta t\theta_1 f(t_m, z(t_m,x), u(t_m,x,.)).\end{aligned} \tag{6.30c}$$

The coefficients $\mathcal{Z}_k(t_m)$, $\mathcal{F}_k(t_m)$, and $\mathcal{Y}_k(t_m)$ are recovered in a similar way, as explained in section 4.4. The computation of the Fourier cosine coefficients $\mathcal{U}_k^\ell(t_m)$ of function $u(t_m,x,j_\ell)$ can be decomposed into three parts. In summary, this results in

$$\begin{aligned}\mathcal{U}_k^\ell(t_m) \approx \Re\Bigg(\sum_{j=0}^{N-1}{}' \Bigg[& -\frac{1-\theta_3}{\theta_3}\mathcal{U}_j^\ell(t_{m+1}) + \frac{1}{\Delta t\theta_3}\Delta t\left[\exp\left(i\frac{k\pi j_\ell}{b-a}\right)-1\right]\mathcal{Y}_j(t_{m+1})\\ &+ \frac{1-\theta_3}{\theta_3}\Delta t\left[\exp\left(i\frac{k\pi j_\ell}{b-a}\right)-1\right]\mathcal{F}_j(t_{m+1})\Bigg]\phi\left(\frac{j\pi}{b-a}\right)\mathcal{M}_{k,j}\Bigg).\end{aligned} \tag{6.31}$$

With the above equations we can recover the Fourier cosine coefficients recursively and solve the BSDEJ backward in time.

6.4. Reference values. We first explain briefly how we can use the COS method, in a completely different way, to obtain reference values for the numerical tests in section 7. The utility maximization problem,

$$V(w) = \max_{\alpha_t\in\mathcal{A}} \mathbb{E}\left[\mathfrak{U}(W_T^\alpha - g(S_T))\right], \tag{6.32}$$

is a two-dimensional (2D) stochastic control problem with the following underlying processes:

$$dS_t/S_{t-} = bdt + \sigma d\omega_t + \int_E \beta(J)\tilde{N}(dJ,dt), \tag{6.33a}$$

$$dW_t^\alpha = \alpha_t bdt + \alpha_t\sigma d\omega_t + \alpha_t\int_E \beta(J)\tilde{N}(dJ,dt). \tag{6.33b}$$

We can solve this problem by dynamic programming and the 2D-COS method. This method was developed in [41] for pricing rainbow options, for which the payoff depends on two or more asset price processes, and can also be applied to stochastic control problems.

If it is not possible to invest in assets and to hedge the risky option, i.e., $\alpha_t = 0$ for all $t \in [0, T]$, then the portfolio value $W_t^\alpha = w$ is constant, and the problem reduces to

$$V(w) = \mathbb{E}\left[\mathfrak{U}(w - g(S_T))\right] = -e^{-\eta w}\mathbb{E}\left[e^{\eta g(S_T)}\right]. \tag{6.34}$$

We can approximate this one-dimensional expectation by using the one-dimensional COS formula from [16].

7. Numerical experiments BSDEJ. In this section we use the BCOS method to value a put option under jump-diffusion asset prices by using utility indifference pricing, as explained in section 6.2. For the numerical tests, we use the following parameter values:

$$S_0 = 1,\ K = 1,\ b = 0.1779,\ \sigma = 0.2,\ T = 0.1. \tag{7.1}$$

The jumps occurring are assumed to be bivariate distributed with possible jump sizes j_1 and j_2, with

$$j_1 = -0.1338,\ j_2 = -0.9838,\ p_1 = p_2 = 0.5,\ \lambda = 0.0228, \tag{7.2}$$

so that the expected value is -0.5588 and the standard deviation is 0.4250. These values correspond to the real-world $\mathbb{P}$-measure for the jump-diffusion asset price in [27].

Similarly to [16], we choose the computational domain

$$[a, b] = \left[x_0 + \kappa_1 - L\sqrt{\kappa_2 + \sqrt{\kappa_4}},\ x_0 + \kappa_1 + L\sqrt{\kappa_2 + \sqrt{\kappa_4}}\right], \quad L = 12. \tag{7.3}$$

The cumulants κ_1, κ_2, and κ_4 of the Brownian motion and the Merton jump-diffusion process are, for example, given in [16]. Again we set the number of terms in the Fourier cosine series expansions equal to $N = 2^9$.

We distinguish between three theta-discretization schemes:

Scheme E:	$\theta_1 = 1,$	$\theta_2 = 1,$	$\theta_3 = 1,$
Scheme F:	$\theta_1 = 0.5,$	$\theta_2 = 1,$	$\theta_3 = 1,$
Scheme G:	$\theta_1 = 0.5,$	$\theta_2 = 0.5,$	$\theta_3 = 0.5.$

For solving (6.26) in the first time iteration, $m = M - 1$, we set $\theta_1 = \theta_2 = \theta_3 = 1$, because the driver function $f(.)$ depends on the unprescribed values $z(t_M, x)$ and $u(t_M, x, .)$.

No hedge. We start with the setting where it is not possible to invest in assets and to hedge the risky option, i.e., $\alpha_t = 0$ for all $t \in [0, T]$. In Figure 4 results of the BCOS method are shown. The left plot shows the initial values of the BSDEs, Y_0^ξ, $Y_0^{-\xi}$, and Y_0^0, for different values of η, and the right plot gives the bid and ask prices. The dots are the values obtained by the BCOS method, while the black circles give the reference value obtained by the COS method as described in section 6.4. The approximated values correspond to the reference values.

Restricted hedging strategy. For the second test we assume that the set of admissible strategies is given by $\mathcal{A} = [-15, 15]$. In other words, a maximum of 15 Euro is used to buy or sell assets. We use Newton's method to find the optimal strategy in (6.8c). Figure 5 presents the results of the BCOS method. The reference values (black circles) are obtained by the 2D-COS method.

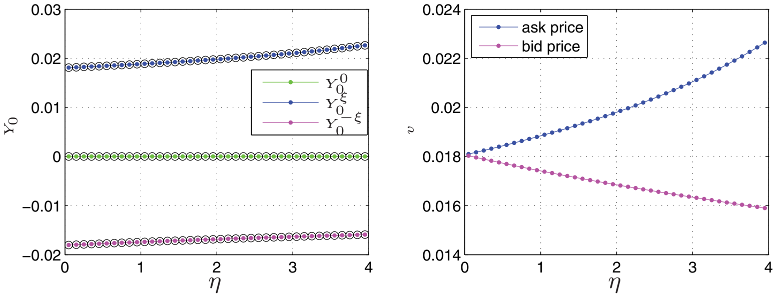

FIG. 4. *Results Y_0 and utility indifference prices (scheme G, $N = 2^9$, $M = 64$, $\Delta t = 0.1/64$). (Color available in electronic version.)*

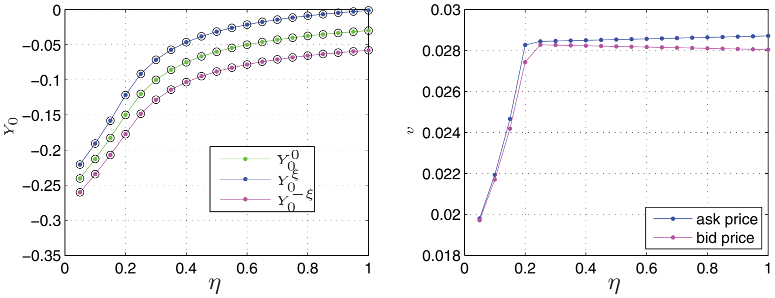

FIG. 5. *Results Y_0 and utility indifference prices (scheme G, $N = 2^9$, $M = 64$, $\Delta t = 0.1/64$). (Color available in electronic version.)*

Convergence in M. For the last test we investigate the convergence of the error in the number of time steps M for $\eta = 1$ and with terminal conditions ξ and $-\xi$. Reference values are obtained by choosing a large number of time steps M. The results are shown in Figure 6. The approximated value $\hat{y}(t_0, x_0)$ converges with $\mathcal{O}(\Delta t)$ for schemes E and F and with $\mathcal{O}((\Delta t)^2)$ for scheme G, as expected. The values $\hat{z}(t_0, x_0)$, $\hat{u}(t_0, x_0, j_1)$, and $\hat{u}(t_0, x_0, j_2)$ converge with $\mathcal{O}(\Delta t)$ for all three schemes. Again the scheme with $\theta_i = 1/2$, $i = 1, 2, 3$, gives the best convergence rate. The CPU times for different values of N and M are shown in Table 2.

8. Conclusions and outlook.

In this paper we proposed a probabilistic numerical method for solving backward stochastic differential equations (BSDEs). The first step consists of discretizing the BSDE by taking conditional expectations and applying a general theta-discretization for the time-integrals. Then, the BCOS method solves the problem backward in time by approximating the conditional expectations with the help of COS formulas. The Fourier cosine coefficients are recovered recursively in an efficient way by using discrete Fourier cosine transforms and an FFT algorithm.

Numerical tests demonstrate the applicability of the BCOS method for BSDEs in economic and financial problems. In the tests we observed different convergence results for Z_0 and Y_0. The convergence of the error in the number of time steps depends on the smoothness and the Lipschitz constant of the driver function and the terminal condition. In general, we achieve the highest convergence rate for the theta-scheme with $\theta_1 = \theta_2 = 1/2$.

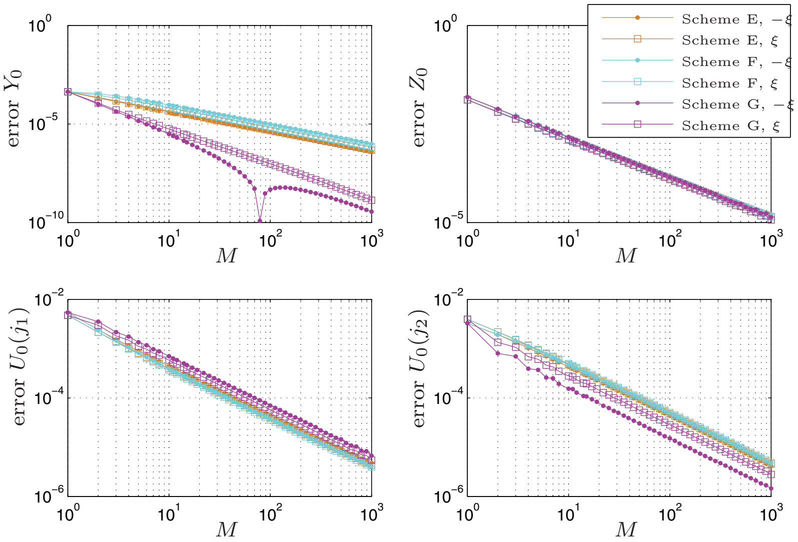

Fig. 6. *Convergence in M ($N = 2^9$). Upper left: Error in $\hat{y}(t_0, x_0)$. Upper right: Error in $\hat{z}(t_0, x_0)$. Lower left: Error in $\hat{u}(t_0, x_0, j_1)$. Lower right: Error in $\hat{u}(t_0, x_0, j_2)$. (Color available in electronic version.)*

Table 2
CPU time (s).

M	4	8	16	32	64	128	256	512
$N = 2^9$	0.0694	0.1086	0.1908	0.3358	0.6428	1.2555	2.4931	4.9387

N	2^6	2^7	2^8	2^9
$M = 256$	0.7897	1.0745	1.5204	2.4931

Utility indifference pricing is used to value options in an incomplete market under a jump-diffusion asset price process, possibly with a restricted hedging portfolio. The bid and ask prices are represented by BSDEs with jumps. We extended our BCOS method to solving these BSDEJs under jump-diffusion with a finite number of jump sizes. Numerical experiments show highly satisfactory and efficient pricing results. The theta-scheme with $\theta_1 = \theta_2 = \theta_3 = 1/2$ gives the fastest convergence.

The COS method is applicable for all Lévy processes and is especially efficient for the affine class. BSDEs driven by Lévy processes are discussed in [35], and they are a challenging extension of the BCOS method. Another interesting extension are second order BSDEs [15], which will be part of our future research.

Appendix A. COS formulas. In this section we explain how to approximate several conditional expectations under the discrete process

$$X^{\Delta}_{m+1} = X^{\Delta}_m + \mu(t_m, x)\Delta t + \sigma(t_m, x)\Delta\omega_{m+1} + \int_E JN(dJ, \Delta t), \tag{A.1}$$

with characteristic function

$$\varphi(u|x) = \varphi(u|0)e^{iux} = \phi(u|x)e^{iux}, \quad \text{with}$$
$$\phi(u|x) := \exp\left(iu\mu(t_m, x)\Delta t - \tfrac{1}{2}u^2\sigma^2(t_m, x)\Delta t\right) e^{\lambda\Delta t(\varphi_J(u)-1)}, \tag{A.2}$$

where $\varphi_J(u) = \sum_{\ell=1}^{n_j} p_\ell e^{iuj_\ell}$ denotes the characteristic function of jump size J.

A.1. Computation of expectation $\mathbb{E}_m^x\left[\cdot\,\Delta\omega_{m+1}\right]$. Integration by parts gives us, for sufficiently smooth v,

$$\begin{aligned}
&\mathbb{E}_m^x\left[v(t_{m+1}, X_{m+1}^\Delta)\Delta\omega_{m+1}\right]\\
&= \mathbb{E}_m^x\left[v\left(t_{m+1}, x+\mu(t_m,x)\Delta t+\sigma(t_m,x)\Delta\omega_{m+1}+\int_E JN(dJ,\Delta t)\right)\Delta\omega_{m+1}\right]\\
&= \mathbb{E}_m^x\left[\frac{1}{\sqrt{2\pi}\sqrt{\Delta t}}\int_{\mathbb{R}} v\left(t_{m+1}, x+\mu(t_m,x)\Delta t+\sigma(t_m,x)\zeta+\int_E JN(dJ,\Delta t)\right)\zeta e^{-\frac{1}{2}\left(\frac{\zeta}{\sqrt{\Delta t}}\right)^2}d\zeta\right]\\
&= \mathbb{E}_m^x\left[\frac{\sigma(t_m,x)\Delta t}{\sqrt{2\pi}\sqrt{\Delta t}}\int_{\mathbb{R}} D_x v\left(t_{m+1}, x+\mu(t_m,x)\Delta t+\sigma(t_m,x)\zeta+\int_E JN(dJ,\Delta t)\right)e^{-\frac{1}{2}\left(\frac{\zeta}{\sqrt{\Delta t}}\right)^2}d\zeta\right]\\
&= \sigma(t_m,x)\Delta t\mathbb{E}_m^x\left[D_x v(t_{m+1}, X_{m+1}^\Delta)\right].
\end{aligned} \tag{A.3}$$

For the error analysis in section 4.5 we assume constant μ and σ terms; then iterated conditioning gives

$$\mathbb{E}_m^x\left[v(t_{m+2}, X_{m+2}^\Delta)\Delta\omega_{m+2}\right] = \sigma\Delta t\mathbb{E}_m^x\left[D_x v(t_{m+2}, X_{m+2}^\Delta)\right] \tag{A.4}$$

and

$$\begin{aligned}
&\mathbb{E}_m^x\left[v(t_{m+2}, X_{m+2}^\Delta)\Delta\omega_{m+1}\right]\\
&\qquad= \mathbb{E}_m^x\left[v\left(t_{m+2}, x+\mu 2\Delta t+\sigma\Delta\omega_{m+1}+\sigma\Delta\omega_{m+2}+\int_E JN(dJ,2\Delta t)\right)\Delta\omega_{m+1}\right]\\
&\qquad= \sigma\Delta t\mathbb{E}_m^x\left[D_x v(t_{m+2}, X_{m+2}^\Delta)\right].
\end{aligned} \tag{A.5}$$

The derivation for diffusion processes can be found by omitting the jump part in the derivation.

A.2. Computation of expectation $\mathbb{E}_m^x\left[\cdot\,\tilde{N}(\{j_\ell\},\Delta t)\right]$. Equation (6.29b) and, similarly, equation (6.29c) require the computation of

$$\begin{aligned}
\mathbb{E}_m^x\left[v(t_{m+1}, X_{m+1}^\Delta)\tilde{N}(\{j_\ell\},\Delta t)\right] &= \mathbb{E}_m^x\left[v(t_{m+1}, X_{m+1}^\Delta)N(\{j_\ell\},\Delta t)\right]\\
&\quad - \mathbb{E}_m^x\left[v(t_{m+1}, X_{m+1}^\Delta)\right]\nu(\{j_\ell\})\Delta t.
\end{aligned} \tag{A.6}$$

The first part in (A.6) can be written as

$$\begin{aligned}
&\mathbb{E}_m^x\left[v(t_{m+1}, X_{m+1}^\Delta)N(\{j_\ell\},\Delta t)\right]\\
&\approx {\sum_{k=0}^{N-1}}' \mathcal{V}_k(t_{m+1})\Re\left(\mathbb{E}_m^x\left[N(\{j_\ell\},\Delta t)\exp\left(iu(X_{m+1}^\Delta-a)\right)\right]\right),
\end{aligned} \tag{A.7}$$

with

$$\begin{aligned}
\mathbb{E}_m^x\left[N(\{j_\ell\},\Delta t)\exp\left(iu(X_{m+1}^\Delta-a)\right)\right] &= \mathbb{E}_m^x\left[\exp\left(iu(x+\mu(t_m,x)\Delta t+\sigma(t_m,x)\Delta\omega_{m+1}-a)\right)\right]\\
&\quad\cdot\mathbb{E}_m^x\left[N(\{j_\ell\},\Delta t)\exp\left(iu\int_E JN(dJ,\Delta t)\right)\right].
\end{aligned} \tag{A.8}$$

Now let $\tau_q, q = 1, 2, \ldots, N_{\Delta t}$, denote the jump times between t_m and t_{m+1}, with jump sizes J_{τ_q}. Then, we find by the law of iterated expectations the following equality:

$$\begin{aligned}
&\mathbb{E}\left[N(\{j_\ell\}, \Delta t)\exp\left(iu\int_E JN(dJ, \Delta t)\right)\right] = \mathbb{E}\left[\sum_{q=1}^{N_{\Delta t}} \mathbf{1}_{j_\ell}(J_{\tau_q})\exp\left(iu\sum_{l=1}^{N_{\Delta t}} J_{\tau_l}\right)\right]\\
&= \mathbb{E}\left[\mathbb{E}\left[\sum_{q=1}^{N_{\Delta t}} \mathbf{1}_{j_\ell}(J_{\tau_q})\exp\left(iu\sum_{l=1}^{N_{\Delta t}} J_{\tau_l}\right)\Big| N_{\Delta t}\right]\right]\\
&= \sum_{n=0}^{\infty} e^{-\lambda\Delta t}\frac{(\lambda\Delta t)^n}{n!}\mathbb{E}\left[\sum_{q=1}^{n} \mathbf{1}_{j_\ell}(J_{\tau_q})\exp\left(iu\sum_{l=1}^{n} J_{\tau_l}\right)\right]\\
&= \sum_{n=0}^{\infty} e^{-\lambda\Delta t}\frac{(\lambda\Delta t)^n}{n!}\sum_{q=1}^{n}\mathbb{E}\left[\mathbf{1}_{j_\ell}(J_{\tau_q})\exp\left(iuJ_{\tau_q}\right)\right]\mathbb{E}\left[\exp\left(iu\sum_{l=1, l\neq q}^{n} J_{\tau_l}\right)\right]\\
&= \sum_{n=0}^{\infty} e^{-\lambda\Delta t}\frac{(\lambda\Delta t)^n}{n!} np_\ell e^{iuj_\ell}\left(\varphi_J(u)\right)^{n-1}\\
&= e^{iuj_\ell}p_\ell\lambda\Delta t e^{\lambda\Delta t(\varphi_J(u)-1)}.
\end{aligned} \tag{A.9}$$

We end up with the approximation

$$\begin{aligned}
&\mathbb{E}_m^x\left[v(t_{m+1}, X_{m+1}^{\Delta})\tilde{N}(\{j_\ell\}, \Delta t)\right]\\
&\approx \sum_{k=0}^{N-1}{}' \mathcal{V}_k(t_{m+1})\Re\left(\phi(u)e^{iu(x-a)}\left[\exp\left(iuj_\ell\right) - 1\right]p_\ell\lambda\Delta t\right).
\end{aligned} \tag{A.10}$$

REFERENCES

[1] G. Barles, R. Buckdahn, and E. Pardoux, *Backward stochastic differential equations and integral-partial differential equations*, Stochastics, 60 (1997), pp. 57–83.

[2] D. Becherer, *Bounded solutions to backward SDEs with jumps for utility optimization and indifference hedging*, Ann. Appl. Probab., 16 (2006), pp. 2027–2054.

[3] C. Bender and R. Denk, *A forward scheme for backward SDEs*, Stochastic Process. Appl., 117 (2007), pp. 1793–1812.

[4] C. Bender and J. Steiner, *Least-squares Monte Carlo for backward SDEs*, Numer. Methods Finance, 12 (2012), pp. 257–289.

[5] J.-M. Bismut, *Conjugate convex functions in optimal stochastic control*, J. Math. Anal. Appl., 44 (1973), pp. 384–404.

[6] F. Black and M. Scholes, *The pricing of options and corporate liabilities*, J. Political Economy, 81 (1973), pp. 637–654.

[7] B. Bouchard and R. Elie, *Discrete-time approximation of decoupled forward–backward SDE with jumps*, Stochastic Process. Appl., 118 (2008), pp. 53–75.

[8] B. Bouchard and N. Touzi, *Discrete-time approximation and Monte-Carlo simulation of backward stochastic differential equations*, Stochastic Process. Appl., 111 (2004), pp. 175–206.

[9] J. P. Boyd, *Chebyshev and Fourier spectral methods*, Courier Dover, New York, 2001.

[10] P. Briand, B. Delyon, and J. Mémin, *Donsker-type theorem for BSDEs*, Electron. Commun. Probab., 6 (2001), pp. 1–14.

[11] P. Briand and C. Labart, *Simulation of BSDEs by Wiener chaos expansion*, HAL: hal-00688523, version 2, 2013; available online from https://hal.inria.fr/hal-00688523v2/document.

[12] D. Crisan and K. Manolarakis, *Solving backward stochastic differential equations using the cubature method: Application to nonlinear pricing*, SIAM J. Financial Math., 3 (2012), pp. 534–571.

[13] N. El Karoui, S. Peng, and M. C. Quenez, *Backward stochastic differential equations in finance*, Math. Finance, 7 (1997), pp. 1–71.

[14] N. El Karoui and M.-C. Quenez, *Dynamic programming and pricing of contingent claims in an incomplete market*, SIAM J. Control Optim., 33 (1995), pp. 29–66.

[15] A. Fahim, N. Touzi, and X. Warin, *A probabilistic numerical method for fully nonlinear parabolic PDEs*, Ann. Appl. Probab., 21 (2011), pp. 1322–1364.

[16] F. Fang and C. W. Oosterlee, *A novel pricing method for European options based on Fourier-cosine series expansions*, SIAM J. Sci. Comput., 31 (2008), pp. 826–848.

[17] F. Fang and C. W. Oosterlee, *Pricing early-exercise and discrete barrier options by Fourier-cosine series expansions*, Numer. Math., 114 (2009), pp. 27–62.

[18] H. Föllmer and D. Sondermann, *Hedging of non-redundant contingent claims*, in Contributions to Mathematical Economics, W. Hildenbrand and A. Mas-Colell, eds., Elsevier Science, North–Holland, Amsterdam, 1986, pp. 205–223.

[19] W. M. Gentleman, *Implementing Clenshaw-Curtis quadrature,* I. *Methodology and experience*, Commun. ACM, 15 (1972), pp. 337–342.

[20] E. Gobet and C. Labart, *Solving BSDE with adaptive control variate*, SIAM J. Numer. Anal., 48 (2010), pp. 257–277.

[21] E. Gobet, J. P. Lemor, and X. Warin, *A regression-based Monte Carlo method to solve backward stochastic differential equations*, Ann. Appl. Probab., 15 (2005), pp. 2172–2202.

[22] E. Gobet and P. Turkedjiev, *Linear regression MDP scheme for discrete backward stochastic differential equations under general conditions*, HAL: hal-00642685, version 3, 2013; available online from https://hal.inria.fr/hal-00642685v3/document.

[23] V. Henderson and D. Hobson, *Utility indifference pricing: An overview*, in Indifference Pricing: Theory and Applications, R. Carmona, ed., Princeton University Press, Princeton, NJ, 2009, pp. 44–74.

[24] S. Hodges and A. Neuberger, *Optimal replication of contingent claims under transactions costs*, Rev. Futures Markets, 8 (1989), pp. 222–239.

[25] Y. Hu, P. Imkeller, and M. Müller, *Utility maximization in incomplete markets*, Ann. Appl. Probab., 15 (2005), pp. 1691–1712.

[26] A. Iserles, *A First Course in the Numerical Analysis of Differential Equations*, 2nd ed., Cambridge Texts in Applied Mathematics, Cambridge University Press, Cambridge, UK, 2009.

[27] J. S. Kennedy, P. A. Forsyth, and K. R. Vetzal, *Dynamic hedging under jump diffusion with transaction costs*, Oper. Res., 57 (2009), pp. 541–559.

[28] M. Kobylanski, *Backward stochastic differential equations and partial differential equations with quadratic growth*, Ann. Probab., 28 (2000), pp. 558–602.

[29] J. P. Lemor, E. Gobet, and X. Warin, *Rate of convergence of an empirical regression method for solving generalized backward stochastic differential equations*, Bernoulli, 12 (2006), pp. 889–916.

[30] J. P. Lepeltier and J. San Martin, *Backward stochastic differential equations with continuous coefficient*, Statist. Probab. Lett., 32 (1997), pp. 425–430.

[31] Y. Li and W. Zhao, *Lp-error estimates for numerical schemes for solving certain kinds of backward stochastic differential equations*, Statist. Probab. Lett., 80 (2010), pp. 1612–1617.

[32] J. Ma, P. Protter, J. San Martín, and S. Torres, *Numerical method for backward stochastic differential equations*, Ann. Appl. Probab., 12 (2002), pp. 302–316.

[33] M. Mania and M. Schweizer, *Dynamic exponential utility indifference valuation*, Ann. Appl. Probab., 15 (2005), pp. 2113–2143.

[34] M. A. Morlais, *A new existence result for quadratic BSDEs with jumps with application to the utility maximization problem*, Stochastic Process. Appl., 120 (2010), pp. 1966–1995.

[35] D. Nualart and W. Schoutens, *Backward stochastic differential equations and Feynman-Kac formula for Lévy processes, with applications in finance*, Bernoulli, 7 (2001), pp. 761–776.

[36] E. Pardoux and S. Peng, *Backward stochastic differential equations and quasilinear parabolic partial differential equations*, in Stochastic Partial Differential Equations and Their Applications, Lecture Notes in Control Inform. Sci. 176., Springer, Berlin, 1992, pp. 200–217.

[37] E. Pardoux and S. G. Peng, *Adapted solution of a backward stochastic differential equation*, Systems Control Lett., 14 (1990), pp. 55–61.

[38] H. Pham, *Continuous-Time Stochastic Control and Optimization with Financial Applications*, Springer-Verlag, Berlin, 2009.

[39] R. Rouge and N. El Karoui, *Pricing via utility maximization and entropy*, Math. Finance,

10 (2000), pp. 259–276.

[40] M. Royer, *Backward stochastic differential equations with jumps and related non-linear expectations*, Stochastic Process. Appl., 116 (2006), pp. 1358–1376.

[41] M. J. Ruijter and C. W. Oosterlee, *Two-dimensional Fourier cosine series expansion method for pricing financial options*, SIAM J. Sci. Comput., 34 (2012), pp. B642–B671.

[42] M. J. Ruijter and C. W. Oosterlee, *Numerical Fourier Method and Second-Order Taylor Scheme for Backward SDEs in Finance*, working paper, SSRN, http://ssrn.com/abstract=2501686, submitted for publication, 2014.

[43] M. J. Ruijter and C. W. Oosterlee, *Supplementary materials: A Fourier cosine method for an efficient computation of solutions to BSDEs*, SIAM J. Sci. Comput., 37 (2015), pp. A859–A889.

[44] M. J. Ruijter, M. Versteegh, and C. W. Oosterlee, *On the application of spectral filters in a Fourier option pricing technique*, J. Comput. Finance, (2015), to appear.

[45] J. Sekine, *On exponential hedging and related quadratic backward stochastic differential equations*, Appl. Math. Optim., 54 (2006), pp. 131–158.

[46] S. Tang and X. Li, *Necessary conditions for optimal control of stochastic systems with random jumps*, SIAM J. Control Optim., 32 (1994), pp. 1447–1475.

[47] J. Wang, C. Luo, and W. Zhao, *Crank-Nicolson scheme and its error estimates for backward stochastic differential equations*, Acta Math. Appl. Sin., November (2009). DOI: 10.1007/s10255-009-9051-z.

[48] J. Zhang, *A numerical scheme for BSDEs*, Ann. Appl. Probab., 14 (2004), pp. 459–488.

[49] W. Zhao, Y. Li, and L. Ju, *Error estimates of the Crank-Nicolson scheme for solving backward stochastic differential equations*, Internat. J. Numer. Anal. Model., 10 (2013), pp. 876–898.

[50] W. Zhao, Y. Li, and Zhang, *A generalized θ-scheme for solving backward stochastic differential equations*, Discrete Contin. Dyn. Syst. Ser. B, 17 (2012), pp. 1585–1603.

[51] W. Zhao, J. Wang, and S. Peng, *Error estimates of the theta-scheme for backward stochastic differential equations*, Discrete Contin. Dyn. Syst. Ser. B, 12 (2009), pp. 905–924.